

LNC5 15302

Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal (Eds.)

Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part II

2 Part II

ICPR
2024 INDIA



 Springer



Lecture Notes in Computer Science

15302

Founding Editors


Gerhard Goos


Juris Hartmanis

Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*

Wen Gao, *Peking University, Beijing, China*

Bernhard Steffen , *TU Dortmund University, Dortmund, Germany*

Moti Yung , *Columbia University, New York, NY, USA*

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.

LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.


Apostolos Antonacopoulos ·
Subhasis Chaudhuri · Rama Chellappa ·
Cheng-Lin Liu · Saumik Bhattacharya ·
Umapada Pal
Editors


Pattern Recognition

27th International Conference, ICPR 2024
Kolkata, India, December 1–5, 2024
Proceedings, Part II

Editors

Apostolos Antonacopoulos 
University of Salford
Salford, UK

Rama Chellappa 
Johns Hopkins University
Baltimore, MD, USA

Saumik Bhattacharya 
IIT Kharagpur
Kharagpur, India

Subhasis Chaudhuri 
Indian Institute of Technology Bombay
Mumbai, India

Cheng-Lin Liu 
Chinese Academy of Sciences
Beijing, China

Umapada Pal 
Indian Statistical Institute Kolkata
Kolkata, India

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-031-78165-0

ISBN 978-3-031-78166-7 (eBook)

<https://doi.org/10.1007/978-3-031-78166-7>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2025

Chapter “Advancing 3D Mesh Analysis: A Graph Learning Approach for Intersecting 3D Geometry Classification” is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). For further details see license information in the chapter.

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

President's Address

On behalf of the Executive Committee of the International Association for Pattern Recognition (IAPR), I am pleased to welcome you to the 27th International Conference on Pattern Recognition (ICPR 2024), the main scientific event of the IAPR.

After a completely digital ICPR in the middle of the COVID pandemic and the first hybrid version in 2022, we can now enjoy a fully back-to-normal ICPR this year. I look forward to hearing inspirational talks and keynotes, catching up with colleagues during the breaks and making new contacts in an informal way. At the same time, the conference landscape has changed. Hybrid meetings have made their entrance and will continue. It is exciting to experience how this will influence the conference. Planning for a major event like ICPR must take place over a period of several years. This means many decisions had to be made under a cloud of uncertainty, adding to the already large effort needed to produce a successful conference. It is with enormous gratitude, then, that we must thank the team of organizers for their hard work, flexibility, and creativity in organizing this ICPR. ICPR always provides a wonderful opportunity for the community to gather together. I can think of no better location than Kolkata to renew the bonds of our international research community.

Each ICPR is a bit different owing to the vision of its organizing committee. For 2024, the conference has six different tracks reflecting major themes in pattern recognition: Artificial Intelligence, Pattern Recognition and Machine Learning; Computer and Robot Vision; Image, Speech, Signal and Video Processing; Biometrics and Human Computer Interaction; Document Analysis and Recognition; and Biomedical Imaging and Bioinformatics. This reflects the richness of our field. ICPR 2024 also features two dozen workshops, seven tutorials, and 15 competitions; there is something for everyone. Many thanks to those who are leading these activities, which together add significant value to attending ICPR, whether in person or virtually. Because it is important for ICPR to be as accessible as possible to colleagues from all around the world, we are pleased that the IAPR, working with the ICPR organizers, is continuing our practice of awarding travel stipends to a number of early-career authors who demonstrate financial need. Last but not least, we are thankful to the Springer LNCS team for their effort to publish these proceedings.

Among the presentations from distinguished keynote speakers, we are looking forward to the three IAPR Prize Lectures at ICPR 2024. This year we honor the achievements of Tin Kam Ho (IBM Research) with the IAPR's most prestigious King-Sun Fu Prize "for pioneering contributions to multi-classifier systems, random decision forests, and data complexity analysis". The King-Sun Fu Prize is given in recognition of an outstanding technical contribution to the field of pattern recognition. It honors the memory of Professor King-Sun Fu who was instrumental in the founding of IAPR, served as its first president, and is widely recognized for his extensive contributions to the field of pattern recognition.

The Maria Petrou Prize is given to a living female scientist/engineer who has made substantial contributions to the field of Pattern Recognition and whose past contributions, current research activity and future potential may be regarded as a model to both aspiring and established researchers. It honours the memory of Professor Maria Petrou as a scientist of the first rank, and particularly her role as a pioneer for women researchers. This year, the Maria Petrou Prize is given to Guoying Zhao (University of Oulu), “for contributions to video analysis for facial micro-behavior recognition and remote biometric reading (RPPG) for heart rate analysis and face anti-spoofing”.

The J.K. Aggarwal Prize is given to a young scientist who has brought a substantial contribution to a field that is relevant to the IAPR community and whose research work has had a major impact on the field. Professor Aggarwal is widely recognized for his extensive contributions to the field of pattern recognition and for his participation in IAPR's activities. This year, the J.K. Aggarwal Prize goes to Xiaolong Wang (UC San Diego) “for groundbreaking contributions to advancing visual representation learning, utilizing self-supervised and attention-based models to establish fundamental frameworks for creating versatile, general-purpose pattern recognition systems”.

During the conference we will also recognize 21 new IAPR Fellows selected from a field of very strong candidates. In addition, a number of Best Scientific Paper and Best Student Paper awards will be presented, along with the Best Industry Related Paper Award and the Piero Zamperoni Best Student Paper Award. Congratulations to the recipients of these very well-deserved awards!

I would like to close by again thanking everyone involved in making ICPR 2024 a tremendous success; your hard work is deeply appreciated. These thanks extend to all who chaired the various aspects of the conference and the associated workshops, my ExCo colleagues, and the IAPR Standing and Technical Committees. Linda O’Gorman, the IAPR Secretariat, deserves special recognition for her experience, historical perspective, and attention to detail when it comes to supporting many of the IAPR’s most important activities. Her tasks became so numerous that she recently got support from Carolyn Buckley (layout, newsletter), Ugur Halici (ICPR matters), and Rosemary Stramka (secretariat). The IAPR website got a completely new design. Ed Sobczak has taken care of our web presence for so many years already. A big thank you to all of you!

This is, of course, the 27th ICPR conference. Knowing that ICPR is organized every two years, and that the first conference in the series (1973!) pre-dated the formal founding of the IAPR by a few years, it is also exciting to consider that we are celebrating over 50 years of ICPR and at the same time approaching the official IAPR 50th anniversary in 2028: you’ll get all information you need at ICPR 2024. In the meantime, I offer my thanks and my best wishes to all who are involved in supporting the IAPR throughout the world.

September 2024

Arjan Kuijper
President of the IAPR

Preface

It is our great pleasure to welcome you to the proceedings of the 27th International Conference on Pattern Recognition (ICPR 2024), held in Kolkata, India. The city, formerly known as ‘Calcutta’, is the home of the fabled Indian Statistical Institute (ISI), which has been at the forefront of statistical pattern recognition for almost a century. Concepts like the Mahalanobis distance, Bhattacharyya bound, Cramer–Rao bound, and Fisher–Rao metric were invented by pioneers associated with ISI. The first ICPR (called IJCPR then) was held in 1973, and the second in 1974. Subsequently, ICPR has been held every other year. The International Association for Pattern Recognition (IAPR) was founded in 1978 and became the sponsor of the ICPR series. Over the past 50 years, ICPR has attracted huge numbers of scientists, engineers and students from all over the world and contributed to advancing research, development and applications in pattern recognition technology.

ICPR 2024 was held at the Biswa Bangla Convention Centre, one of the largest such facilities in South Asia, situated just 7 kilometers from Kolkata Airport (CCU). According to ChatGPT “Kolkata is often called the ‘Cultural Capital of India’. The city has a deep connection to literature, music, theater, and art. It was home to Nobel laureate Rabindranath Tagore, and the Bengali film industry has produced globally renowned filmmakers like Satyajit Ray. The city boasts remarkable colonial architecture, with landmarks like Victoria Memorial, Howrah Bridge, and the Indian Museum (the oldest and largest museum in India). Kolkata’s streets are dotted with old mansions and buildings that tell stories of its colonial past. Walking through the city can feel like stepping back into a different era. Finally, Kolkata is also known for its street food.”

ICPR 2024 followed a two-round paper submission format. We received a total of 2135 papers (1501 papers in round-1 submissions, and 634 papers in round-2 submissions). Each paper, on average, received 2.84 reviews, in single-blind mode. For the first-round papers we had a rebuttal option available to authors.

In total, 945 papers (669 from round-1 and 276 from round-2) were accepted for presentation, resulting in an acceptance rate of 44.26%, which is consistent with previous ICPR events. At ICPR 2024 the papers were categorized into six tracks: Artificial Intelligence, Machine Learning for Pattern Analysis; Computer Vision and Robotic Perception; Image, Video, Speech, and Signal Analysis; Biometrics and Human-Machine Interaction; Document and Media Analysis; and Biomedical Image Analysis and Informatics.

The main conference ran over December 2–5, 2024. The main program included the presentation of 188 oral papers (19.89% of the accepted papers), 757 poster papers and 12 competition papers (out of 15 submitted). A total 10 oral sessions were held concurrently in four meeting rooms with a total of 40 oral sessions. In total 24 workshops and 7 tutorials were held on December 1, 2024.

The plenary sessions included three prize lectures and three invited presentations. The prize lectures were delivered by Tin Kam Ho (IBM Research, USA; King Sun

Fu Prize winner), Xiaolong Wang (University of California, San Diego, USA; J.K. Aggarwal Prize winner), and Guoying Zhao (University of Oulu, Finland; Maria Petrou Prize winner). The invited speakers were Timothy Hospedales (University of Edinburgh, UK), Venu Govindaraju (University at Buffalo, USA), and Shuicheng Yan (Skywork AI, Singapore).

Several best paper awards were presented in ICPR: the Piero Zamperoni Award for the best paper authored by a student, the BIRPA Best Industry Related Paper Award, and the Best Paper Awards and Best Student Paper Awards for each of the six tracks of ICPR 2024.

The organization of such a large conference would not be possible without the help of many volunteers. Our special gratitude goes to the Program Chairs (Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa and Cheng-Lin Liu), for their leadership in organizing the program. Thanks to our Publication Chairs (Ananda S. Chowdhury and Wataru Ohyama) for handling the overwhelming workload of publishing the conference proceedings. We also thank our Competition Chairs (Richard Zanibbi, Lianwen Jin and Laurence Likforman-Sulem) for arranging 12 important competitions as part of ICPR 2024. We are thankful to our Workshop Chairs (P. Shivakumara, Stephanie Schuckers, Jean-Marc Ogier and Prabir Bhattacharya) and Tutorial Chairs (B.B. Chaudhuri, Michael R. Jenkin and Guoying Zhao) for arranging the workshops and tutorials on emerging topics. ICPR 2024, for the first time, held a Doctoral Consortium. We would like to thank our Doctoral Consortium Chairs (Véronique Eglin, Dan Lopresti and Mayank Vatsa) for organizing it.

Thanks go to the Track Chairs and the meta reviewers who devoted significant time to the review process and preparation of the program. We also sincerely thank the reviewers who provided valuable feedback to the authors.

Finally, we acknowledge the work of other conference committee members, like the Organizing Chairs and Organizing Committee Members, Finance Chairs, Award Chair, Sponsorship Chairs, and Exhibition and Demonstration Chairs, Visa Chair, Publicity Chairs, and Women in ICPR Chairs, whose efforts made this event successful. We also thank our event manager Alpcord Network for their help.

We hope that all the participants found the technical program informative and enjoyed the sights, culture and cuisine of Kolkata.

October 2024

Umapada Pal
Josef Kittler
Anil Jain

Organization

General Chairs

Umapada Pal
Josef Kittler
Anil Jain

Indian Statistical Institute, Kolkata, India
University of Surrey, UK
Michigan State University, USA

Program Chairs

Apostolos Antonacopoulos
Subhasis Chaudhuri
Rama Chellappa
Cheng-Lin Liu

University of Salford, UK
Indian Institute of Technology, Bombay, India
Johns Hopkins University, USA
Institute of Automation, Chinese Academy of
Sciences, China

Publication Chairs

Ananda S. Chowdhury
Wataru Ohyama

Jadavpur University, India
Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi
Lianwen Jin
Laurence Likforman-Sulem

Rochester Institute of Technology, USA
South China University of Technology, China
Télécom Paris, France

Workshop Chairs

P. Shivakumara
Stephanie Schuckers
Jean-Marc Ogier
Prabir Bhattacharya

University of Salford, UK
Clarkson University, USA
Université de la Rochelle, France
Concordia University, Canada

Tutorial Chairs

B. B. Chaudhuri	Indian Statistical Institute, Kolkata, India
Michael R. Jenkin	York University, Canada
Guoying Zhao	University of Oulu, Finland

Doctoral Consortium Chairs

Véronique Eglin	CNRS, France
Daniel P. Lopresti	Lehigh University, USA
Mayank Vatsa	Indian Institute of Technology, Jodhpur, India

Organizing Chairs

Saumik Bhattacharya	Indian Institute of Technology, Kharagpur, India
Palash Ghosal	Sikkim Manipal University, India

Organizing Committee

Santanu Phadikar	West Bengal University of Technology, India
SK Md Obaidullah	Aliah University, India
Sayantari Ghosh	National Institute of Technology Durgapur, India
Himadri Mukherjee	West Bengal State University, India
Nilamadhaba Tripathy	Clarivate Analytics, USA
Chayan Halder	West Bengal State University, India
Shibaprasad Sen	Techno Main Salt Lake, India

Finance Chairs

Kaushik Roy	West Bengal State University, India
Michael Blumenstein	University of Technology Sydney, Australia

Awards Committee Chair

Arpan Pal	Tata Consultancy Services, India
-----------	----------------------------------

Sponsorship Chairs

P. J. Narayanan	Indian Institute of Technology, Hyderabad, India
Yasushi Yagi	Osaka University, Japan
Venu Govindaraju	University at Buffalo, USA
Alberto Bel Bimbo	Università di Firenze, Italy

Exhibition and Demonstration Chairs

Arjun Jain	FastCode AI, India
Agnimitra Biswas	National Institute of Technology, Silchar, India

International Liaison, Visa Chair

Balasubramanian Raman	Indian Institute of Technology, Roorkee, India
-----------------------	--

Publicity Chairs

Dipti Prasad Mukherjee	Indian Statistical Institute, Kolkata, India
Bob Fisher	University of Edinburgh, UK
Xiaojun Wu	Jiangnan University, China

Women in ICPR Chairs

Ingela Nystrom	Uppsala University, Sweden
Alexandra B. Albu	University of Victoria, Canada
Jing Dong	Institute of Automation, Chinese Academy of Sciences, China
Sarbani Palit	Indian Statistical Institute, Kolkata, India

Event Manager

Alpcord Network

Track Chairs – Artificial Intelligence, Machine Learning for Pattern Analysis

Larry O’Gorman	Nokia Bell Labs, USA
Dacheng Tao	University of Sydney, Australia
Petia Radeva	University of Barcelona, Spain
Susmita Mitra	Indian Statistical Institute, Kolkata, India
Jiliang Tang	Michigan State University, USA

Track Chairs – Computer and Robot Vision

C. V. Jawahar	International Institute of Information Technology (IIIT), Hyderabad, India
João Paulo Papa	São Paulo State University, Brazil
Maja Pantic	Imperial College London, UK
Gang Hua	Dolby Laboratories, USA
Junwei Han	Northwestern Polytechnical University, China

Track Chairs – Image, Speech, Signal and Video Processing

P. K. Biswas	Indian Institute of Technology, Kharagpur, India
Shang-Hong Lai	National Tsing Hua University, Taiwan
Hugo Jair Escalante	INAOE, CINVESTAV, Mexico
Sergio Escalera	Universitat de Barcelona, Spain
Prem Natarajan	University of Southern California, USA

Track Chairs – Biometrics and Human Computer Interaction

Richa Singh	Indian Institute of Technology, Jodhpur, India
Massimo Tistarelli	University of Sassari, Italy
Vishal Patel	Johns Hopkins University, USA
Wei-Shi Zheng	Sun Yat-sen University, China
Jian Wang	Snap, USA

Track Chairs – Document Analysis and Recognition

Xiang Bai	Huazhong University of Science and Technology, China
David Doermann	University at Buffalo, USA
Josep Lladós	Universitat Autònoma de Barcelona, Spain
Mita Nasipuri	Jadavpur University, India

Track Chairs – Biomedical Imaging and Bioinformatics

Jayanta Mukhopadhyay	Indian Institute of Technology, Kharagpur, India
Xiaoyi Jiang	Universität Münster, Germany
Seong-Whan Lee	Korea University, Korea

Metareviewers (Conference Papers and Competition Papers)

Wael Abd-Elmageed	University of Southern California, USA
Maya Aghaei	NHL Stenden University, Netherlands
Alireza Alaei	Southern Cross University, Australia
Rajagopalan N. Ambasamudram	Indian Institute of Technology, Madras, India
Suyash P. Awate	Indian Institute of Technology, Bombay, India
Inci M. Baytas	Bogazici University, Turkey
Aparna Bharati	Lehigh University, USA
Brojeshwar Bhowmick	Tata Consultancy Services, India
Jean-Christophe Burie	University of La Rochelle, France
Gustavo Carneiro	University of Surrey, UK
Chee Seng Chan	Universiti Malaya, Malaysia
Sumohana S. Channappayya	Indian Institute of Technology, Hyderabad, India
Dongdong Chen	Microsoft, USA
Shengyong Chen	Tianjin University of Technology, China
Jun Cheng	Institute for Infocomm Research, A*STAR, Singapore
Albert Clapés	University of Barcelona, Spain
Oscar Dalmau	Center for Research in Mathematics, Mexico

Tyler Derr	Vanderbilt University, USA
Abhinav Dhall	Indian Institute of Technology, Ropar, India
Bo Du	Wuhan University, China
Yuxuan Du	University of Sydney, Australia
Ayman S. El-Baz	University of Louisville, USA
Francisco Escolano	University of Alicante, Spain
Siamac Fazli	Nazarbayev University, Kazakhstan
Jianjiang Feng	Tsinghua University, China
Gernot A. Fink	TU Dortmund University, Germany
Alicia Fornes	CVC, Spain
Junbin Gao	University of Sydney, Australia
Yan Gao	Amazon, USA
Yongsheng Gao	Griffith University, Australia
Caren Han	University of Melbourne, Australia
Ran He	Institute of Automation, Chinese Academy of Sciences, China
Tin Kam Ho	IBM, USA
Di Huang	Beihang University, China
Kaizhu Huang	Duke Kunshan University, China
Donato Impedovo	University of Bari, Italy
Julio Jacques	University of Barcelona and Computer Vision Center, Spain
Lianwen Jin	South China University of Technology, China
Wei Jin	Emory University, USA
Danilo Samuel Jodas	São Paulo State University, Brazil
Manjunath V. Joshi	DA-IICT, India
Jayashree Kalpathy-Cramer	Massachusetts General Hospital, USA
Dimosthenis Karatzas	Computer Vision Centre, Spain
Hamid Karimi	Utah State University, USA
Baiying Lei	Shenzhen University, China
Guoqi Li	Chinese Academy of Sciences, and Peng Cheng Lab, China
Laurence Likforman-Sulem	Institut Polytechnique de Paris/Télécom Paris, France
Aishan Liu	Beihang University, China
Bo Liu	Bytedance, USA
Chen Liu	Clarkson University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China
Hongmin Liu	University of Science and Technology Beijing, China
Hui Liu	Michigan State University, USA

Jing Liu	Institute of Automation, Chinese Academy of Sciences, China
Li Liu	University of Oulu, Finland
Qingshan Liu	Nanjing University of Posts and Telecommunications, China
Adrian P. Lopez-Monroy	Centro de Investigacion en Matematicas AC, Mexico
Daniel P. Lopresti	Lehigh University, USA
Shijian Lu	Nanyang Technological University, Singapore
Yong Luo	Wuhan University, China
Andreas K. Maier	FAU Erlangen-Nuremberg, Germany
Davide Maltoni	University of Bologna, Italy
Hong Man	Stevens Institute of Technology, USA
Lingtong Min	Northwestern Polytechnical University, China
Paolo Napoletano	University of Milano-Bicocca, Italy
Kamal Nasrollahi	Milestone Systems, Aalborg University, Denmark
Marcos Ortega	University of A Coruña, Spain
Shivakumara Palaiahnakote	University of Salford, UK
P. Jonathon Phillips	NIST, USA
Filiberto Pla	University Jaume I, Spain
Ajit Rajwade	Indian Institute of Technology, Bombay, India
Shanmuganathan Raman	Indian Institute of Technology, Gandhinagar, India
Imran Razzak	UNSW, Australia
Beatriz Remeseiro	University of Oviedo, Spain
Gustavo Rohde	University of Virginia, USA
Partha Pratim Roy	Indian Institute of Technology, Roorkee, India
Sanjoy K. Saha	Jadavpur University, India
Joan Andreu Sánchez	Universitat Politècnica de València, Spain
Claudio F. Santos	UFSCar, Brazil
Shin'ichi Satoh	National Institute of Informatics, Japan
Stephanie Schuckers	Clarkson University, USA
Srirangaraj Setlur	University at Buffalo, SUNY, USA
Debdoot Sheet	Indian Institute of Technology, Kharagpur, India
Jun Shen	University of Wollongong, Australia
Li Shen	JD Explore Academy, China
Chen Shengyong	Zhejiang University of Technology and Tianjin University of Technology, China
Andy Song	RMIT University, Australia
Akihiro Sugimoto	National Institute of Informatics, Japan
Qianru Sun	Singapore Management University, Singapore
Arijit Sur	Indian Institute of Technology, Guwahati, India
Estefania Talavera	University of Twente, Netherlands

Wei Tang	University of Illinois at Chicago, USA
Joao M. Tavares	Universidade do Porto, Portugal
Jun Wan	NLPR, CASIA, China
Le Wang	Xi'an Jiaotong University, China
Lei Wang	Australian National University, Australia
Xiaoyang Wang	Tencent AI Lab, USA
Xinggang Wang	Huazhong University of Science and Technology, China
Xiao-Jun Wu	Jiangnan University, China
Yiding Yang	Bytedance, China
Xiwen Yao	Northwestern Polytechnical University, China
Xu-Cheng Yin	University of Science and Technology Beijing, China
Baosheng Yu	University of Sydney, Australia
Shiqi Yu	Southern University of Science and Technology, China
Xin Yuan	Westlake University, China
Yibing Zhan	JD Explore Academy, China
Jing Zhang	University of Sydney, Australia
Lefei Zhang	Wuhan University, China
Min-Ling Zhang	Southeast University, China
Wenbin Zhang	Florida International University, USA
Jiahuan Zhou	Peking University, China
Sanping Zhou	Xi'an Jiaotong University, China
Tianyi Zhou	University of Maryland, USA
Lei Zhu	Shandong Normal University, China
Pengfei Zhu	Tianjin University, China
Wangmeng Zuo	Harbin Institute of Technology, China

Reviewers (Competition Papers)

Liangcai Gao	Da-Han Wang
Mingxin Huang	Yang Xue
Lei Kang	Wentao Yang
Wenhui Liao	Jiaxin Zhang
Yuliang Liu	Yiwu Zhong
Yongxin Shi	

Reviewers (Conference Papers)

Aakanksha Aakanksha
 Aayush Singla
 Abdul Muqet
 Abhay Yadav
 Abhijeet Vijay Nandedkar
 Abhimanyu Sahu
 Abhinav Rajvanshi
 Abhisek Ray
 Abhishek Shrivastava
 Abhra Chaudhuri
 Aditi Roy
 Adriano Simonetto
 Adrien Maglo
 Ahmed Abdulkadir
 Ahmed Boudissa
 Ahmed Hamdi
 Ahmed Rida Sekkat
 Ahmed Sharafeldeen
 Aiman Farooq
 Aishwarya Venkataramanan
 Ajay Kumar
 Ajay Kumar Reddy Poreddy
 Ajita Rattani
 Ajoy Mondal
 Akbar K.
 Akbar Telikani
 Akshay Agarwal
 Akshit Jindal
 Al Zadid Sultan Bin Habib
 Albert Clapés
 Alceu Britto
 Alejandro Peña
 Alessandro Ortis
 Alessia Auriemma Citarella
 Alexandre Stenger
 Alexandros Sopasakis
 Alexia Toumpa
 Ali Khan
 Alik Pramanick
 Alireza Alaei
 Alper Yilmaz
 Aman Verma
 Amit Bhardwaj

Amit More
 Amit Nandedkar
 Amitava Chatterjee
 Amos L. Abbott
 Amrita Mohan
 Anand Mishra
 Ananda S. Chowdhury
 Anastasia Zakharova
 Anastasios L. Kesidis
 Andras Horvath
 Andre Gustavo Hochuli
 André P. Kelm
 Andre Wyzykowski
 Andrea Bottino
 Andrea Lagorio
 Andrea Torsello
 Andreas Fischer
 Andreas K. Maier
 Andreu Girbau Xalabarder
 Andrew Beng Jin Teoh
 Andrew Shin
 Andy J. Ma
 Aneesh S. Chivukula
 Ángela Casado-García
 Anh Quoc Nguyen
 Anindya Sen
 Anirban Saha
 Anjali Gautam
 Ankan Bhattacharyya
 Ankit Jha
 Anna Scius-Bertrand
 Annalisa Franco
 Antoine Doucet
 Antonino Staiano
 Antonio Fernández
 Antonio Parziale
 Anu Singha
 Anustup Choudhury
 Anwesha Pal
 Anwesha Sengupta
 Archisman Adhikary
 Arjan Kuijper
 Arnab Kumar Das

Arnav Bhavsar	Bin-Bin Jia
Arnav Varma	Binbin Yong
Arpita Dutta	Bindita Chaudhuri
Arshad Jamal	Bindu Madhavi Tummala
Artur Jordao	Binh M. Le
Arunkumar Chinnaswamy	Bi-Ru Dai
Aryan Jadon	Bo Huang
Aryaz Baradarani	Bo Jiang
Ashima Anand	Bob Zhang
Ashis Dhara	Bowen Liu
Ashish Phophalia	Bowen Zhang
Ashok K. Bhateja	Boyang Zhang
Ashutosh Vaish	Boyu Diao
Ashwani Kumar	Boyun Li
Asifuzzaman Lasker	Brian M. Sadler
Atefeh Khoshkhahtinat	Bruce A. Maxwell
Athira Nambiar	Bryan Bo Cao
Attilio Fiandrotti	Buddhika L. Semage
Avandra S. Hemachandra	Bushra Jalil
Avik Hati	Byeong-Seok Shin
Avinash Sharma	Byung-Gyu Kim
B. H. Shekar	Caihua Liu
B. Uma Shankar	Cairong Zhao
Bala Krishna Thunakala	Camille Kurtz
Balaji Tk	Carlos A. Caetano
Balázs Pálffy	Carlos D. Martá-Nez-Hinarejos
Banafsheh Adami	Ce Wang
Bang-Dang Pham	Cevahir Cigla
Baochang Zhang	Chakravarthy Bhagvati
Baodi Liu	Chandrakanth Vipparla
Bashirul Azam Biswas	Changchun Zhang
Beiduo Chen	Changde Du
Benedikt Kottler	Changkun Ye
Beomseok Oh	Changxu Cheng
Berkay Aydin	Chao Fan
Berlin S. Shaheema	Chao Guo
Bertrand Kerautret	Chao Qu
Bettina Finzel	Chao Wen
Bhavana Singh	Chayan Halder
Bibhas C. Dhara	Che-Jui Chang
Bilge Günsel	Chen Feng
Bin Chen	Chenan Wang
Bin Li	Cheng Yu
Bin Liu	Chenghao Qian
Bin Yao	Cheng-Lin Liu

Chengxu Liu
Chenru Jiang
Chensheng Peng
Chetan Ralekar
Chih-Wei Lin
Chih-Yi Chiu
Chinmay Sahu
Chintan Patel
Chintan Shah
Chiranjoy Chattopadhyay
Chong Wang
Choudhary Shyam Prakash
Christophe Charrier
Christos Smailis
Chuanwei Zhou
Chun-Ming Tsai
Chunpeng Wang
Ciro Russo
Claudio De Stefano
Claudio F. Santos
Claudio Marrocco
Connor Levenson
Constantine Dovrolis
Constantine Kotropoulos
Dai Shi
Dakshina Ranjan Kisku
Dan Anitei
Dandan Zhu
Daniela Pamplona
Danli Wang
Danqing Huang
Daoan Zhang
Daqing Hou
David A. Clausi
David Freire Obregon
David Münch
David Pujol Perich
Davide Marelli
De Zhang
Debalina Barik
Debapriya Roy (Kundu)
Debashis Das
Debashis Das Chakladar
Debi Prosad Dogra
Debraj D. Basu

Decheng Liu
Deen Dayal Mohan
Deep A. Patel
Deepak Kumar
Dengpan Liu
Denis Coquenat
Désiré Sidibé
Devesh Walawalkar
Dewan Md. Farid
Di Ming
Di Qiu
Di Yuan
Dian Jia
Dianmo Sheng
Diego Thomas
Diganta Saha
Dimitri Bulatov
Dimpy Varshni
Dingcheng Yang
Dipanjan Das
Dipanjoy Paul
Divya Biligere Shivanna
Divya Saxena
Divya Sharma
Dmitrii Matveichev
Dmitry Minskiy
Dmitry V. Sorokin
Dong Zhang
Donghua Wang
Donglin Zhang
Dongming Wu
Dongqiangzi Ye
Dongqing Zou
Dongrui Liu
Dongyang Zhang
Dongzhan Zhou
Douglas Rodrigues
Duarte Folgado
Duc Minh Vo
Duoxuan Pei
Durai Arun Pannir Selvam
Durga Bhavani S.
Eckart Michaelsen
Elena Goyanes
Élodie Puybareau

Emanuele Vivoli	Galal Binamakhshen
Emna Ghorbel	Ganesh Krishnasamy
Enrique Naredo	Gang Pan
Enyu Cai	Gangyan Zeng
Eric Patterson	Gani Rahmon
Ernest Valveny	Gaurav Harit
Eva Blanco-Mallo	Gennaro Vessio
Eva Breznik	Genoveffa Tortora
Evangelos Sartinias	George Azzopardi
Fabio Solari	Gerard Ortega
Fabiola De Marco	Gerardo E. Altamirano-Gomez
Fan Wang	Gernot A. Fink
Fangda Li	Gibran Benitez-Garcia
Fangyuan Lei	Gil Ben-Artzi
Fangzhou Lin	Gilbert Lim
Fangzhou Luo	Giorgia Minello
Fares Bougourzi	Giorgio Fumera
Farman Ali	Giovanna Castellano
Fatiha Mokdad	Giovanni Puglisi
Fei Shen	Giulia Orrù
Fei Teng	Giuliana Ramella
Fei Zhu	Gökçe Uludoğan
Feiyan Hu	Gopi Ramena
Felipe Gomes Oliveira	Gorthi Rama Krishna Sai Subrahmanyam
Feng Li	Gourav Datta
Fengbei Liu	Gowri Srinivasa
Fenghua Zhu	Gozde Sahin
Fillipe D. M. De Souza	Gregory Randall
Flavio Piccoli	Guanjie Huang
Flavio Prieto	Guanjun Li
Florian Kleber	Guanwen Zhang
Francesc Serratosa	Guanyu Xu
Francesco Bianconi	Guanyu Yang
Francesco Castro	Guanzhou Ke
Francesco Ponzio	Guhnoo Yun
Francisco Javier Hernández López	Guido Borghi
Frédéric Rayar	Guilherme Brandão Martins
Furkan Osman Kar	Guillaume Caron
Fushuo Huo	Guillaume Tochon
Fuxiao Liu	Guocai Du
Fu-Zhao Ou	Guohao Li
Gabriel Turinici	Guoqiang Zhong
Gabrielle Flood	Guorong Li
Gajjala Viswanatha Reddy	Guotao Li
Gaku Nakano	Gurman Gill

Haechang Lee
Haichao Zhang
Haidong Xie
Haifeng Zhao
Haimei Zhao
Hainan Cui
Haixia Wang
Haiyan Guo
Hakime Ozturk
Hamid Kazemi
Han Gao
Hang Zou
Hanjia Lyu
Hanjoo Cho
Hanqing Zhao
Hanyuan Liu
Hanzhou Wu
Hao Li
Hao Meng
Hao Sun
Hao Wang
Hao Xing
Hao Zhao
Haoan Feng
Haodi Feng
Haofeng Li
Haoji Hu
Haojie Hao
Haojun Ai
Haopeng Zhang
Haoran Li
Haoran Wang
Haorui Ji
Haoxiang Ma
Haoyu Chen
Haoyue Shi
Harald Koestler
Harbinder Singh
Harris V. Georgiou
Hasan F. Ates
Hasan S. M. Al-Khaffaf
Hatef Otroshi Shahreza
Hebeizi Li
Heng Zhang
Hengli Wang

Hengyue Liu
Hertog Nugroho
Hieyong Jeong
Himadri Mukherjee
Hoai Ngo
Hoda Mohaghegh
Hong Liu
Hong Man
Hongcheng Wang
Hongjian Zhan
Hongxi Wei
Hongyu Hu
Hoseong Kim
Hossein Ebrahimnezhad
Hossein Malekmohamadi
Hrishav Bakul Barua
Hsueh-Yi Sean Lin
Hua Wei
Huafeng Li
Huali Xu
Huaming Chen
Huan Wang
Huang Chen
Huanran Chen
Hua-Wen Chang
Huawen Liu
Huayi Zhan
Hugo Jair Escalante
Hui Chen
Hui Li
Huichen Yang
Huiqiang Jiang
Huiyuan Yang
Huizi Yu
Hung T. Nguyen
Hyeongyu Kim
Hyeonjeong Park
Hyeonjun Lee
Hymalai Bello
Hyung-Gun Chi
Hyunsoo Kim
I-Chen Lin
Ik Hyun Lee
Ilan Shimshoni
Imad Eddine Toubal

Imran Sarker
Inderjot Singh Saggu
Indrani Mukherjee
Indranil Sur
Ines Rieger
Ioannis Pierros
Irina Rabaev
Ivan V. Medri
J. Rafid Siddiqui
Jacek Komorowski
Jacopo Bonato
Jacson Rodrigues Correia-Silva
Jaekoo Lee
Jaime Cardoso
Jakob Gawlikowski
Jakub Nalepa
James L. Wayman
Jan Čech
Jangho Lee
Jani Boutellier
Javier Gurrola-Ramos
Javier Lorenzo-Navarro
Jayasree Saha
Jean Lee
Jean Paul Barddal
Jean-Bernard Hayet
Jean-Philippe G. Tarel
Jean-Yves Ramel
Jenny Benois-Pineau
Jens Bayer
Jerin Geo James
Jesús Miguel García-Gorrostieta
Jia Qu
Jiahong Chen
Jiaji Wang
Jian Hou
Jian Liang
Jian Xu
Jian Zhu
Jianfeng Lu
Jianfeng Ren
Jiangfan Liu
Jianguo Wang
Jiangyan Yi
Jiangyong Duan

Jianhua Yang
Jianhua Zhang
Jianhui Chen
Jianjia Wang
Jianli Xiao
Jianqiang Xiao
Jianwu Wang
Jianxin Zhang
Jianxiong Gao
Jianxiong Zhou
Jianyu Wang
Jianzhong Wang
Jiaru Zhang
Jiashu Liao
Jiaxin Chen
Jiaxin Lu
Jiaxing Ye
Jiaxuan Chen
Jiaxuan Li
Jiayi He
Jiayin Lin
Jie Ou
Jiehua Zhang
Jiejie Zhao
Jignesh S. Bhatt
Jin Gao
Jin Hou
Jin Hu
Jin Shang
Jing Tian
Jing Yu Chen
Jingfeng Yao
Jinglun Feng
Jingtong Yue
Jingwei Guo
Jingwen Xu
Jingyuan Xia
Jingzhe Ma
Jinhong Wang
Jinjia Wang
Jinlai Zhang
Jinlong Fan
Jinming Su
Jinrong He
Jintao Huang

Jinwoo Ahn
Jinwoo Choi
Jinyang Liu
Jinyu Tian
Jionghao Lin
Jiuding Duan
Jiwei Shen
Jiyang Pan
Jiyoun Kim
João Papa
Johan Debayle
John Atanbori
John Wilson
John Zhang
Jónathan Heras
Joohi Chauhan
Jorge Calvo-Zaragoza
Jorge Figueroa
Jorma Laaksonen
José Joaquim De Moura Ramos
Jose Vicent
Joseph Damilola Akinyemi
Josiane Zerubia
Juan Wen
Judit Szücs
Juepeng Zheng
Juha Roning
Jumana H. Alsubhi
Jun Cheng
Jun Ni
Jun Wan
Junghyun Cho
Junjie Liang
Junjie Ye
Junlin Hu
Juntong Ni
Junxin Lu
Junxuan Li
Junyaup Kim
Junyeong Kim
Jürgen Seiler
Jushang Qiu
Juyang Weng
Jyostna Devi Bodapati
Jyoti Singh Kirar

Kai Jiang
Kaiqiang Song
Kalidas Yeturu
Kalle Åström
Kamalakar Vijay Thakare
Kang Gu
Kang Ma
Kanji Tanaka
Karthik Seemakurthy
Kaushik Roy
Kavisha Jayathunge
Kazuki Uehara
Ke Shi
Keigo Kimura
Keiji Yanai
Kelton A. P. Costa
Kenneth Camilleri
Kenny Davila
Ketan Atul Bapat
Ketan Kotwal
Kevin Desai
Keyu Long
Khadiga Mohamed Ali
Khakon Das
Khan Muhammad
Kilho Son
Kim-Ngan Nguyen
Kishan Kc
Kishor P. Upla
Klaas Dijkstra
Komal Bharti
Konstantinos Triaridis
Kostas Ioannidis
Koyel Ghosh
Kripabandhu Ghosh
Krishnendu Ghosh
Kshitij S. Jadhav
Kuan Yan
Kun Ding
Kun Xia
Kun Zeng
Kunal Banerjee
Kunal Biswas
Kunchi Li
Kurban Ubul

Lahiru N. Wijayasingha	Longteng Guo
Laines Schmalwasser	Lorena Álvarez Rodríguez
Lakshman Mahto	Lorenzo Putzu
Lala Shakti Swarup Ray	Lu Leng
Lale Akarun	Lu Pang
Lan Yan	Lu Wang
Lawrence Amadi	Luan Pham
Lee Kang Il	Luc Brun
Lei Fan	Luca Guarnera
Lei Shi	Luca Piano
Lei Wang	Lucas Alexandre Ramos
Leonardo Rossi	Lucas Goncalves
Lequan Lin	Lucas M. Gago
Levente Tamas	Luigi Celona
Li Bing	Luis C. S. Afonso
Li Li	Luis Gerardo De La Fraga
Li Ma	Luis S. Luevano
Li Song	Luis Teixeira
Lia Morra	Lunke Fei
Liang Xie	M. Hassaballah
Liang Zhao	Maddimsetti Srinivas
Lianwen Jin	Mahendran N.
Libing Zeng	Mahesh Mohan M. R.
Lidia Sánchez-González	Maiko Lie
Lidong Zeng	Mainak Singha
Lijun Li	Makoto Hirose
Likang Wang	Malay Bhattacharyya
Lili Zhao	Mamadou Dian Bah
Lin Chen	Man Yao
Lin Huang	Manali J. Patel
Linfei Wang	Manav Prabhakar
Ling Lo	Manikandan V. M.
Lingchen Meng	Manish Bhatt
Lingheng Meng	Manjunath Shantharamu
Lingxiao Li	Manuel Curado
Lingzhong Fan	Manuel Günther
Liqi Yan	Manuel Marques
Liqiang Jing	Marc A. Kastner
Lisa Gutzeit	Marc Chaumont
Liu Ziyi	Marc Cheong
Liushuai Shi	Marc Lalonde
Liviu-Daniel Stefan	Marco Cotogni
Liyuan Ma	Marcos C. Santana
Liyun Zhu	Mario Molinara
Lizuo Jin	Mariofanna Milanova

Markus Bauer
Marlon Becker
Mårten Wadenbäck
Martin G. Ljungqvist
Martin Kämpel
Martina Pastorino
Marwan Torki
Masashi Nishiyama
Masayuki Tanaka
Massimo O. Spata
Matteo Ferrara
Matthew D. Dawkins
Matthew Gadd
Matthew S. Watson
Maura Pintor
Max Ehrlich
Maxim Popov
Mayukh Das
Md Baharul Islam
Md Sajid
Meghna Kapoor
Meghna P. Ayyar
Mei Wang
Meiqi Wu
Melissa L. Tijink
Meng Li
Meng Liu
Meng-Luen Wu
Mengnan Liu
Mengxi China Guo
Mengya Han
Michaël Clément
Michal Kawulok
Mickael Coustaty
Miguel Domingo
Milind G. Padalkar
Ming Liu
Ming Ma
Mingchen Feng
Mingde Yao
Minghao Li
Mingjie Sun
Ming-Kuang Daniel Wu
Mingle Xu
Mingyong Li

Mingyuan Jiu
Minh P. Nguyen
Minh Q. Tran
Minheng Ni
Minsu Kim
Minyi Zhao
Mirko Paolo Barbato
Mo Zhou
Modesto Castrillón-Santana
Mohamed Amine Mezghich
Mohamed Dahmane
Mohamed Elsharkawy
Mohamed Yousuf
Mohammad Hashemi
Mohammad Khalooei
Mohammad Khateri
Mohammad Mahdi Dehshibi
Mohammad Sadil Khan
Mohammed Mahmoud
Moises Diaz
Monalisha Mahapatra
Monidipa Das
Mostafa Kamali Tabrizi
Mridul Ghosh
Mrinal Kanti Bhowmik
Muchao Ye
Mugalodi Ramesha Rakesh
Muhammad Rameez Ur Rahman
Muhammad Suhaib Kanroo
Muming Zhao
Munender Varshney
Munsif Ali
Na Lv
Nader Karimi
Nagabhushan Somraj
Nakkwan Choi
Nakul Agarwal
Nan Pu
Nan Zhou
Nancy Mehta
Nand Kumar Yadav
Nandakishor Nandakishor
Nandyala Hemachandra
Nanfeng Jiang
Narayan Hegde

Narayan Ji Mishra
Narayan Vetrekar
Narendra D. Londhe
Nathalie Girard
Nati Ofir
Naval Kishore Mehta
Nazmul Shahadat
Neeti Narayan
Neha Bhargava
Nemanja Djuric
Newlin Shebiah R.
Ngo Ba Hung
Nhat-Tan Bui
Niaz Ahmad
Nick Theisen
Nicolas Passat
Nicolas Ragot
Nicolas Sidere
Nikolaos Mitianoudis
Nikolas Ebert
Nilah Ravi Nair
Nilesh A. Ahuja
Nilkanta Sahu
Nils Murrugarra-Llerena
Nina S. T. Hirata
Ninad Aithal
Ning Xu
Ningzhi Wang
Niraj Kumar
Nirmal S. Punjabi
Nisha Varghese
Norio Tagawa
Obaidullah Md Sk
Oguzhan Ulucan
Olfa Mechi
Oliver Tüselmann
Orazio Pontorno
Oriol Ramos Terrades
Osman Akin
Ouadi Beya
Ozge Mercanoglu Sincan
Pabitra Mitra
Padmanabha Reddy Y. C. A.
Palaash Agrawal
Palaiahnakote Shivakumara

Palash Ghosal
Pallav Dutta
Paolo Rota
Paramanand Chandramouli
Paria Mehrani
Parth Agrawal
Partha Basuchowdhuri
Patrick Horain
Pavan Kumar
Pavan Kumar Anasosalu Vasu
Pedro Castro
Peipei Li
Peipei Yang
Peisong Shen
Peiyu Li
Peng Li
Pengfei He
Pengrui Quan
Pengxin Zeng
Pengyu Yan
Peter Eisert
Petra Gomez-Krämer
Pierrick Bruneau
Ping Cao
Pingping Zhang
Pintu Kumar
Pooja Kumari
Pooja Sahani
Prabhu Prasad Dev
Pradeep Kumar
Pradeep Singh
Pranjal Sahu
Prasun Roy
Prateek Keserwani
Prateek Mittal
Praveen Kumar Chandaliya
Praveen Tirupattur
Pravin Nair
Preeti Gopal
Preety Singh
Prem Shanker Yadav
Prerana Mukherjee
Prerna A. Mishra
Prianka Dey
Priyanka Mudgal

Qc Kha Ng
 Qi Li
 Qi Ming
 Qi Wang
 Qi Zuo
 Qian Li
 Qiang Gan
 Qiang He
 Qiang Wu
 Qiangqiang Zhou
 Qianli Zhao
 Qiansen Hong
 Qiao Wang
 Qidong Huang
 Qihua Dong
 Qin Yuke
 Qing Guo
 Qingbei Guo
 Qingchao Zhang
 Qingjie Liu
 Qinhong Yang
 Qiushi Shi
 Qixiang Chen
 Quan Gan
 Quanlong Guan
 Rachit Chhaya
 Radu Tudor Ionescu
 Rafal Zdunek
 Raghavendra Ramachandra
 Rahimul I. Mazumdar
 Rahul Kumar Ray
 Rajib Dutta
 Rajib Ghosh
 Rakesh Kumar
 Rakesh Paul
 Rama Chellappa
 Rami O. Skaik
 Ramon Aranda
 Ran Wei
 Ranga Raju Vatsavai
 Ranganath Krishnan
 Rasha Friji
 Rashmi S.
 Razaib Tariq
 Rémi Giraud

René Schuster
 Renlong Hang
 Renrong Shao
 Renu Sharma
 Reza Sadeghian
 Richard Zanibbi
 Rimon Elias
 Rishabh Shukla
 Rita Delussu
 Riya Verma
 Robert J. Ravier
 Robert Sablatnig
 Robin Strand
 Rocco Pietrini
 Rocío Diaz Martin
 Rocío Gonzalez-Diaz
 Rohit Venkata Sai Dulam
 Romain Giot
 Romi Banerjee
 Ru Wang
 Ruben Machucho
 Ruddy Théodose
 Ruggero Pintus
 Rui Deng
 Rui P. Paiva
 Rui Zhao
 Ruifan Li
 Ruigang Fu
 Ruikun Li
 Ruirui Li
 Ruixiang Jiang
 Ruowei Jiang
 Rushi Lan
 Rustam Zhumagambetov
 S. Amutha
 S. Divakar Bhat
 Sagar Goyal
 Sahar Siddiqui
 Sahbi Bahroun
 Sai Karthikeya Vemuri
 Saibal Dutta
 Saihui Hou
 Sajad Ahmad Rather
 Saksham Aggarwal
 Sakthi U.

Salimeh Sekeh	Shinichiro Omachi
Samar Bouazizi	Shirley David
Samia Boukir	Shishir Shah
Samir F. Harb	Shiv Ram Dubey
Samit Biswas	Shiva Baghel
Samrat Mukhopadhyay	Shivanand S. Gornale
Samriddha Sanyal	Shogo Sato
Sandika Biswas	Shotaro Miwa
Sandip Purnapatra	Shreya Ghosh
Sanghyun Jo	Shreya Goyal
Sangwoo Cho	Shuai Su
Sanjay Kumar	Shuai Wang
Sankaran Iyer	Shuai Zheng
Sanket Biswas	Shuaifeng Zhi
Santanu Roy	Shuang Qiu
Santosh D. Pandure	Shuhei Tarashima
Santosh Ku Behera	Shujing Lyu
Santosh Nanabhau Palaskar	Shuliang Wang
Santosh Prakash Chouhan	Shun Zhang
Sarah S. Alotaibi	Shunming Li
Sasanka Katreddi	Shunxin Wang
Sathyanarayanan N. Aakur	Shuping Zhao
Saurabh Yadav	Shuquan Ye
Sayan Rakshit	Shuwei Huo
Scott McCloskey	Shuyue Lan
Sebastian Bunda	Shyi-Chyi Cheng
Sejuti Rahman	Si Chen
Selim Aksoy	Siddarth Ravichandran
Sen Wang	Sihan Chen
Seraj A. Mostafa	Siladittya Manna
Shanmuganathan Raman	Silambarasan Elkana Ebinazer
Shao-Yuan Lo	Simon Benaïchouche
Shaoyuan Xu	Simon S. Woo
Sharia Arfin Tanim	Simone Caldarella
Shehreen Azad	Simone Milani
Sheng Wan	Simone Zini
Shengdong Zhang	Sina Lotfian
Shengwei Qin	Sitao Luan
Shenyuan Gao	Sivaselvan B.
Sherry X. Chen	Siwei Li
Shibaprasad Sen	Siwei Wang
Shigeaki Namiki	Siwen Luo
Shiguang Liu	Siyu Chen
Shijie Ma	Sk Aziz Ali
Shikun Li	Sk Md Obaidullah

Sneha Shukla
Snehasis Banerjee
Snehasis Mukherjee
Snigdha Sen
Sofia Casarin
Soheila Farokhi
Soma Bandyopadhyay
Son Minh Nguyen
Son Xuan Ha
Sonal Kumar
Sonam Gupta
Sonam Nahar
Song Ouyang
Sotiris Kotsiantis
Souhaila Djaffal
Soumen Biswas
Soumen Sinha
Soumitri Chattopadhyay
Souvik Sengupta
Spiros Kostopoulos
Sreeraj Ramachandran
Sreya Banerjee
Srikanta Pal
Srinivas Arukonda
Stephane A. Guinard
Su O. Ruan
Subhadip Basu
Subhajit Paul
Subhankar Ghosh
Subhankar Mishra
Subhankar Roy
Subhash Chandra Pal
Subhayu Ghosh
Sudip Das
Sudipta Banerjee
Suhas Pillai
Sujit Das
Sukalpa Chanda
Sukhendu Das
Suklav Ghosh
Suman K. Ghosh
Suman Samui
Sumit Mishra
Sungho Suh
Sunny Gupta

Suraj Kumar Pandey
Surendrabikram Thapa
Suresh Sundaram
Sushil Bhattacharjee
Susmita Ghosh
Swakkhar Shatabda
Syed Ms Islam
Syed Tousiful Haque
Taegyeong Lee
Taihui Li
Takashi Shibata
Takeshi Oishi
Talha Ahmad Siddiqui
Tanguy Gernot
Tangwen Qian
Tanima Bhowmik
Tanpia Tasnim
Tao Dai
Tao Hu
Tao Sun
Taoran Yi
Tapan Shah
Taveena Lotey
Teng Huang
Tengqi Ye
Teresa Alarcon
Tetsuji Ogawa
Thanh Phuong Nguyen
Thanh Tuan Nguyen
Thattapon Surasak
Thibault Napoléon
Thierry Bouwmans
Thinh Truong Huynh Nguyen
Thomas De Min
Thomas E. K. Zielke
Thomas Swearingen
Tianatahina Jimmy Francky Randrianasoa
Tianheng Cheng
Tianjiao He
Tianyi Wei
Tianyuan Zhang
Tianyue Zheng
Tiecheng Song
Tilottama Goswami
Tim Büchner

Tim H. Langer	Wataru Ohyama
Tim Raven	Wee Kheng Leow
Tingkai Liu	Wei Chen
Tingting Yao	Wei Cheng
Tobias Meisen	Wei Hua
Toby P. Breckon	Wei Lu
Tong Chen	Wei Pan
Tonghua Su	Wei Tian
Tran Tuan Anh	Wei Wang
Tri-Cong Pham	Wei Wei
Trishna Saikia	Wei Zhou
Trung Quang Truong	Weidi Liu
Tuan T. Nguyen	Weidong Yang
Tuan Vo Van	Weijun Tan
Tushar Shinde	Weimin Lyu
Ujjwal Karn	Weinan Guan
Ukrit Watchareeruetai	Weining Wang
Uma Mudenagudi	Weiqliang Wang
Umarani Jayaraman	Weiwei Guo
V. S. Malemath	Weixia Zhang
Vallidevi Krishnamurthy	Wei-Xuan Bao
Ved Prakash	Weizhong Jiang
Venkata Krishna Kishore Kolli	Wen Xie
Venkata R. Vavilthota	Wenbin Qian
Venkatesh Thirugnana Sambandham	Wenbin Tian
Verónica Maria Vasconcelos	Wenbin Wang
Véronique Ve Eglin	Wenbo Zheng
Víctor E. Alonso-Pérez	Wenhan Luo
Vinay Palakkode	Wenhao Wang
Vinayak S. Nageli	Wen-Hung Liao
Vincent J. Whannou De Dravo	Wenjie Li
Vincenzo Conti	Wenkui Yang
Vincenzo Gattulli	Wenwen Si
Vineet Padmanabhan	Wenwen Yu
Vishakha Pareek	Wenwen Zhang
Viswanath Gopalakrishnan	Wenwu Yang
Vivek Singh Baghel	Wenxi Li
Vivekraj K.	Wenxi Yue
Vladimir V. Arlazarov	Wenxue Cui
Vu-Hoang Tran	Wenzhuo Liu
W. Sylvia Lilly Jebarani	Widhiyo Sudiyono
Wachirawit Ponghiran	Willem Dijkstra
Wafa Khlif	Wolfgang Fuhl
Wang An-Zhi	Xi Zhang
Wanli Xue	Xia Yuan

Xianda Zhang	Xuchen Li
Xiang Zhang	Xuejing Lei
Xiangdong Su	Xuelu Feng
Xiang-Ru Yu	Xueting Liu
Xiangtai Li	Xuwei Li
Xiangyu Xu	Xueyi X. Wang
Xiao Guo	Xugong Qin
Xiao Hu	Xu-Qian Fan
Xiao Wu	Xuxu Liu
Xiao Yang	Xu-Yao Zhang
Xiaofeng Zhang	Yan Huang
Xiaogang Du	Yan Li
Xiaoguang Zhao	Yan Wang
Xiaoheng Jiang	Yan Xia
Xiaohong Zhang	Yan Zhuang
Xiaohua Huang	Yanan Li
Xiaohua Li	Yanan Zhang
Xiao-Hui Li	Yang Hou
Xiaolong Sun	Yang Jiao
Xiaosong Li	Yang Liping
Xiaotian Li	Yang Liu
Xiaoting Wu	Yang Qian
Xiaotong Luo	Yang Yang
Xiaoyan Li	Yang Zhao
Xiaoyang Kang	Yangbin Chen
Xiaoyi Dong	Yangfan Zhou
Xin Guo	Yanhui Guo
Xin Lin	Yanjia Huang
Xin Ma	YanJun Zhu
Xinchi Zhou	Yanming Zhang
Xingguang Zhang	Yanqing Shen
Xingjian Leng	Yaoming Cai
Xingpeng Zhang	Yaoxin Zhuo
Xingzheng Lyu	Yaoyan Zheng
Xinjian Huang	Yaping Zhang
Xinqi Fan	Yaqian Liang
Xinqi Liu	Yarong Feng
Xinqiao Zhang	Yasmina Benmabrouk
Xinrui Cui	Yasufumi Sakai
Xizhan Gao	Yasutomo Kawanishi
Xu Cao	Yazeed Alzahrani
Xu Ouyang	Ye Du
Xu Zhao	Ye Duan
Xuan Shen	Yechao Zhang
Xuan Zhou	Yeong-Jun Cho

Yi Huo	Yuchong Yao
Yi Shi	Yuecong Min
Yi Yu	Yuewei Yang
Yi Zhang	Yufei Zhang
Yibo Liu	Yufeng Yin
Yibo Wang	Yugen Yi
Yi-Chieh Wu	Yuhang Ming
Yifan Chen	Yujia Zhang
Yifei Huang	Yujun Ma
Yihao Ding	Yukiko Kenmochi
Yijie Tang	Yun Hoyeoung
Yikun Bai	Yun Liu
Yimin Wen	Yunhe Feng
Yinan Yang	Yunxiao Shi
Yin-Dong Zheng	Yuru Wang
Yinfeng Yu	Yushun Tang
Ying Dai	Yusuf Osmanlioglu
Yingbo Li	Yusuke Fujita
Yiqiao Li	Yuta Nakashima
Yiqing Huang	Yuwei Yang
Yisheng Lv	Yuwu Lu
Yisong Xiao	Yuxi Liu
Yite Wang	Yuya Obinata
Yizhe Li	Yuyao Yan
Yong Wang	Yuzhi Guo
Yonghao Dong	Zaipeng Xie
Yong-Hyuk Moon	Zander W. Blasingame
Yongjie Li	Zedong Wang
Yongqian Li	Zeliang Zhang
Yongqiang Mao	Zexin Ji
Yongxu Liu	Zhanxiang Feng
Yongyu Wang	Zhaofei Yu
Yongzhi Li	Zhe Chen
Youngha Hwang	Zhe Cui
Yousri Kessentini	Zhe Liu
Yu Wang	Zhe Wang
Yu Zhou	Zhekun Luo
Yuan Tian	Zhen Yang
Yuan Zhang	Zhenbo Li
Yuanbo Wen	Zhenchun Lei
Yuanxin Wang	Zhenfei Zhang
Yubin Hu	Zheng Liu
Yubo Huang	Zheng Wang
Yuchen Ren	Zhengming Yu
Yucheng Xing	Zhengyin Du

Zhengyun Cheng
 Zhenshen Qu
 Zhenwei Shi
 Zhenzhong Kuang
 Zhi Cai
 Zhi Chen
 Zhibo Chu
 Zhicun Yin
 Zhida Huang
 Zhida Zhang
 Zhifan Gao
 Zhihang Ren
 Zhihang Yuan
 Zhihao Wang
 Zhihua Xie
 Zhihui Wang
 Zhikang Zhang
 Zhiming Zou
 Zhiqi Shao
 Zhiwei Dong
 Zhiwei Qi
 Zhixiang Wang
 Zhixuan Li
 Zhiyu Jiang
 Zhiyuan Yan
 Zhiyuan Yu
 Zhiyuan Zhang
 Zhong Chen

Zhongwei Teng
 Zhongzhan Huang
 Zhongzhi Yu
 Zhuang Han
 Zhuangzhuang Chen
 Zhuo Liu
 Zhuo Su
 Zhuojun Zou
 Zhuoyue Wang
 Ziang Song
 Zicheng Zhang
 Zied Mnasri
 Zifan Chen
 Žiga Babnik
 Zijing Chen
 Zikai Zhang
 Ziling Huang
 Zilong Du
 Ziqi Cai
 Ziqi Zhou
 Zi-Rui Wang
 Zirui Zhou
 Ziwen He
 Ziyao Zeng
 Ziyi Zhang
 Ziyue Xiang
 Zonglei Jing
 Zongyi Xu

Contents – Part II

Adversarial Transport Terms for Unsupervised Domain Adaptation	1
<i>P. Chirag, Mukta Wagle, Ravi Kant Gupta, Jeevan P. Pranav, and Amit Sethi</i>	
FedSOKD-TFA: Federated Learning with Stage-Optimal Knowledge Distillation and Three-Factor Aggregation	17
<i>Jianhao Liu, Wenjuan Gong, Tingbo Shi, Kechen Li, Yi Jin, and Jordi González</i>	
DualViT: A Hierarchical Vision Transformer for Broad and Fine Class Embeddings	35
<i>Ankita Chatterjee, Sandip Dutta, Jayanta Mukhopadhyay, and Partha Pratim Das</i>	
Establishing Interconnections of Similarity-Based Classifiers for Multi-label Learning with Missing Labels	50
<i>Sambhav Jain and Reshma Rastogi</i>	
GL-TSVM: A Robust and Smooth Twin Support Vector Machine with Guardian Loss Function	63
<i>Mushir Akhtar, M. Tanveer, and Mohd. Arshad</i>	
An Approach Towards Learning K-Means-Friendly Deep Latent Representation	79
<i>Debapriya Roy</i>	
PulmoNetX: A Hybrid Vision Transformer Approach for Multi-scale Spatial Feature Reduction in Pneumonia Classification	93
<i>Asifuzzaman Lasker, Mridul Ghosh, Sk Md Obaidullah, Chandan Chakraborty, Kaushik Roy, and Umapada Pal</i>	
Federated K-Means Clustering	107
<i>Swier Garst and Marcel Reinders</i>	
Feature Selection Voting Strategies and Hyperparameter Tuning in a Boosting Classification	123
<i>Nicole Dalia Cilia, Giovanni Fanara, and Vincenzo Conti</i>	

Advancing 3D Mesh Analysis: A Graph Learning Approach for Intersecting 3D Geometry Classification	143
<i>Stefan Andreas Böhm, Martin Neumayer, Bare Luka Zagar, Fabian Riß, Christian Kortüm, and Alois Knoll</i>	
Efficient Classification of Histopathology Images Using Highly Imbalanced Data	160
<i>Mohammad Iqbal Nouyed, Mary-Anne Hartley, Gianfranco Doretto, and Donald A. Adjeroh</i>	
GenFormer – Generated Images Are All You Need to Improve Robustness of Transformers on Small Datasets	176
<i>Sven Oehri, Nikolas Ebert, Ahmed Abdullah, Didier Stricker, and Oliver Wasenmüller</i>	
Recognizing Patterns of Parkinson’s Disease Using Online Trail Making Test and Response Dynamics – Preliminary Study	193
<i>Artur Chudzik, Jerzy P. Nowacki, and Andrzej W. Przybyszewski</i>	
Regularization of Interpolation Kernel Machines	208
<i>Jiaqi Zhang and Xiaoyi Jiang</i>	
Task Success Classification with Final State of Future Prediction for Robot Control Planning	224
<i>Taku Fujitomi, Naoya Sogi, Takashi Shibata, and Makoto Terao</i>	
EGOFALLS: A Visual-Audio Dataset and Benchmark for Fall Detection Using Egocentric Cameras	240
<i>Xueyi Wang</i>	
Towards Unbiased Minimal Cluster Analysis of Categorical-and-Numerical Attribute Data	254
<i>Yunfan Zhang, Xiaopeng Luo, Qingsheng Chen, Rong Zou, Yiqun Zhang, and Yiu-ming Cheung</i>	
PolSAR Image Classification Using Complex-Valued Squeeze and Excitation Network	270
<i>Shradha Makhija, Srimanta Mandal, Utkarsh Pandya, Sanid Chirakkal, and Deepak Putrevu</i>	
Probabilistic Fusion Framework Combining CNNs and Graphical Models for Multiresolution Satellite and UAV Image Classification	287
<i>Martina Pastorino, Gabriele Moser, Fabien Guerra, Sebastiano B. Serpico, and Josiane Zerubia</i>	

Multiscale Color Guided Attention Ensemble Classifier for Age-Related Macular Degeneration Using Concurrent Fundus and Optical Coherence Tomography Images	304
<i>Pragya Gupta, Subhamoy Mandal, Debashree Guha, and Debjani Chakraborty</i>	
PolSAR Image Classification Using Superpixel Profile and CNN	320
<i>Nabajyoti Das, Swarnajyoti Patra, and Amos Bortiew</i>	
Know How Much Sensitive Precision and Recall Validity Measures Are?	335
<i>Sumit Mishra, Srinibas Swain, and Ved Prakash</i>	
Optimizing Software Release Management with GPT-Enabled Log Anomaly Detection	351
<i>Jahanggir Hossain Setu, Md. Shazzad Hossain, Nabarun Halder, Ashraful Islam, and M. Ashraful Amin</i>	
Patch-Based Prototypical Cross-Scale Attention Network for Anomaly Detection	366
<i>Tung-Lin Wang, Jun-Wei Hsieh, and Yi-Kuan Hsieh</i>	
Semi-structured Pruning of Graph Convolutional Networks for Skeleton-Based Recognition	382
<i>Hichem Sahbi</i>	
Data Pruning via Separability, Integrity, and Model Uncertainty-Aware Importance Sampling	398
<i>Steven Grosz, Rui Zhao, Rajeev Ranjan, Hongcheng Wang, Manoj Aggarwal, Gerard Medioni, and Anil Jain</i>	
Label-Specific Multi-label Classification with Entropy Guided Clustering	414
<i>Jiaxuan Li, Tong Zhu, Xiaoyan Zhu, and Jiayin Wang</i>	
FAT-LSTM: A Multimodal Data Fusion Model with Gating and Attention-Based LSTM for Time-Series Classification	430
<i>Pouya Hosseinzadeh, Omar Bahri, Soukaina Filali Boubrahimi, and Shah Muhammad Hamdi</i>	
Fusing Image and Text Features for Scene Sentiment Analysis Using Whale-Honey Badger Optimization Algorithm (WHBOA)	446
<i>Prem Shanker Yadav, Dinesh Kumar Tyagi, and Santosh Kumar Vipparthi</i>	

EncodeNet: A Framework for Boosting DNN Accuracy
with Entropy-Driven Generalized Converting Autoencoder 463
 Hasanul Mahmud, Palden Lama, Kevin Desai, and Sushil K. Prasad

Author Index 479



Adversarial Transport Terms for Unsupervised Domain Adaptation

P. Chirag, Mukta Wagle, Ravi Kant Gupta^(✉), Jeevan P. Pranav,
and Amit Sethi

Indian Institute of Technology Bombay, Powai, Mumbai, India
{18B090003,18D070054,184070025,194070025,asethi}@iitb.ac.in

Abstract. We propose a new technique called ATT: Adversarial Transport Terms for Unsupervised Domain Adaptation. Adversarial training is commonly used for learning domain-invariant representations by reversing the gradients from a domain discriminator head to train the feature extractor layers of a neural network. We propose significant modifications to the adversarial head, its training objective, and the classifier head. With the aim of reducing class confusion, we introduce a sub-network which displaces the classifier outputs of the source and target domain samples in a learnable manner. We control this movement using a novel transport loss that spreads class clusters away from each other and makes it easier for the classifier to find the decision boundaries for both the source and target domains. The results of adding this new loss to a careful selection of previously proposed losses leads to improvement in UDA results compared to the previous state-of-the-art methods on benchmark datasets. We show the importance of the proposed loss term using ablation studies and visualization of the movement of target domain sample in representation space.

Keywords: ATT · Adversarial · Transport · Unsupervised Domain Adaptation

1 Introduction

Domain shift is a practical problem faced by machine learning models in the real-world. When the distribution of the test data (roughly, *target* domain) is different from that of the main component of the training data (roughly, *source* domain), it causes a drop in model generalization accuracy on the former. Domain adaptation (DA) is a sub-field of machine learning that focuses on developing algorithms and techniques to train models that effectively transfer knowledge learned from a source domain to a target domain [17]. The DA problem may be cast as one of learning to represent the source and the target domain samples in common feature space where their distributions overlap and the subsequent part of the model is able to perform the task with good generalization on both domains [11]. DA techniques can be further classified based on various assumptions about

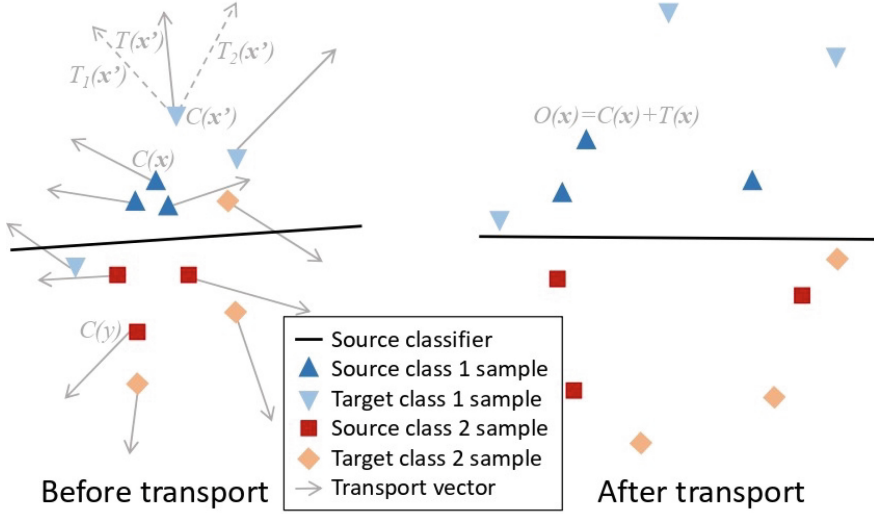


Fig. 1. Unsupervised Domain Adaptation for classification can be achieved by shifting the samples in a way that favors finding a common decision boundary for source and target domains. See text for details

the extent of availability of data from the target domain during training. For instance, semi-supervised DA assumes availability of some labeled samples and lots of unlabeled samples, few-shot DA assumes some labeled samples and no unlabeled samples, unsupervised DA (UDA) assumes only unlabeled samples, and domain generalization assumes that no sample is available from the target domain during training. For a classification task, if the classes in the source and target domains are the same, it is called closed-set domain adaptation. We provide a novel solution for UDA for the closed-set. UDA is instrumental in training transferable models which are pre-trained on a different dataset, without having to worry about labeling any number of images from the target domain. It has major applications in various areas, including medical image analysis, where the image acquisition setup (e.g., equipment, technician training and protocols) is often different between model development and deployment scenarios.

There are some area-specific UDA techniques, such as color normalization for pathology images [26], but we are interested in a more general solution. A common approach for a general UDA technique is to use a moment matching algorithm. It minimizes the absolute difference in moments of the source and target distributions, and aligns feature distributions [12, 29]. This approach has its limitations since it does not account for semantically different features that may have similar moments. Another set of more successful approaches to UDA have been those that train networks to extract feature that confuse a domain discriminator neural network in an adversarial fashion. Such neural networks extract domain-invariant representations, i.e., those cannot be used to distinguish between examples from the source and target domains [5, 16]. These methods have achieved success due to their generative nature, which involves

creating synthetic data to align distributions, as well as their straightforward training process. However, the generation of data points can lead to congestion in the classification space near the decision boundary. To address this, we propose a novel scheme to *transport* samples in the classification (logit) space for adversarial UDA (see Fig. 1).

CAT, or Coupled Adversarial Transport for Unsupervised Domain Adaptation, is our proposed approach. Our architecture is adversarial by nature [15]. We shift each final representation of the source and target samples by two learnable and sample-dependent displacement vectors, which we call *transport* terms. We train a feature extractor in a way that the two transport terms displace different samples in different directions. We also take care that the two transport terms move similar samples similarly. This is done by a bilinear function that we call the transport loss, which has the class information optionally embedded into the loss matrix.

We evaluate the performance of CAT on Office-Home [28], Office-31 [20], and a medical image dataset – FHIST [22]. Medical image classification is a critical yet challenging application of deep domain adaptation as image acquisition setups often change. Class confusion is a rampant problem in medical image datasets, owing to the near identical features of images of different classes [9]. We posit that reducing class confusion in medical image domain adaptation tasks is crucial, and CAT when optionally combined with class confusion minimization solves this problem well.

Our contributions include reducing classification confusion by spatially repositioning samples and formulating a loss term to direct similar samples similarly. We also design an efficient end-to-end network that effectively combines source and target datasets with novel and existing loss functions, and provide a detailed analysis of the impacts of transport and class confusion losses. This approach not only simplifies computational demands compared to complex Domain Adaptation architectures but also offers robust performance across the OfficeHome [28], Office-31 [20], and FHIST datasets [22] in the task of UDA.

2 Background and Related Works

In UDA, we have the source domain samples along with their labels $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ where the number of classes is C . We also have the unlabeled target domain samples $D_t = \{x_i^t\}_{i=1}^{N_t}$. Most contemporary UDA techniques train neural networks end-to-end with multiple loss functions, instead of training multiple networks separately [2, 13, 16, 23, 24, 29]. One group of loss functions for end-to-end training tries to improve classification accuracy on the labeled samples (source domain). An example of such losses is the cross entropy, given as:

$$\mathcal{L}_c = \frac{1}{N_s} \sum_{i=1}^{N_s} L_{ce}(C(G(x_i^s)), y_i^s), \quad (1)$$

where G is a feature extractor sub-network, C is a classifier sub-network, and L_{ce} is the cross entropy loss. Another group of losses tries to match the

distributions of representations of the source and target domain samples. An example of this is Maximum Mean Discrepancy (MMD) [29]. Additional loss terms may be used to keep unlabeled samples (target domain) well-separated near classification boundaries, our major contribution lies here.

Moment matching methods, such as MMD [29], MDD [12] and DCAN [13], aim to match expectations (means) and higher moments of the features of samples of the source and the target domains. However, these techniques do not solve the subtle problem of feature confusion, where one feature is confused for the another between the two domains. For example, the second moments of two features that are semantically different between the two domains could be equal. Consequently, correlation alignment would align incompatible features. Deep CORAL [23] and HoMM [2] propose matching additional measures of the distributions, such as the covariance matrix, to solve this problem.

Adversarial methods take another approach that aims to confuse an additional network on top of the feature extractor, called the discriminator, whose aim is to discriminate between the two domains. The confusion is formulated in terms of an adversarial loss. The two representative methods of this class are DANN [5] and CDAN [16]. CDAN is an improved version of DANN because it captures the relationships between the parameters of the feature extractor and the classifier, rather than just treating them independently. SRDC uses a clustering algorithm to group similar data points together, and finally optimizes the deep representation learning using a joint objective that combines clustering loss and domain classification loss [24].

There are also certain hybrid techniques, which use a combination of moment matching and adversarial training concepts. SHOT projects the data onto a set of random directions, computes the 1D Wasserstein distances between the projected distributions, and optimizes the projections to minimize the Wasserstein distance [14]. SymNets uses three classifiers, one specifically for the source domain, one specifically for the target domain and one for both, and seeks to minimize classification loss on all of them [31]. CyCADA maximizes the cycle consistency which tracks the feature distribution of the image and enforces that the distributions after a cyclical transform are equal [7]. GVB-G is a theoretically sound method, which directly minimizes the extent of the discrepancies by minimizing the length of certain bridge terms between the two domains [3].

Some methods also aim to make the classifier more robust in order to learn invariant feature representations. Maximum Classifier Discrepancy (MCD) aims to learn different representations of the source and target, and learning is made more robust by increasing the discrepancies between the two classifiers in terms of the features they use to classify [21]. Minimum Class Confusion (MCC) aids the classifier to make more confident predictions by minimizing the class confusion probabilities, and in doing so, reducing the number of samples being represented very close to the decision boundaries in the classification space [9].

3 Proposed Approach: ATT

As mentioned in the previous section, the two objectives of several successful UDA methods are to improve the classification accuracy on labeled samples (source domain) and to increase the overlap between the feature distributions of the two domains.

Our proposed approach to UDA pivots on addressing the challenge posed by ambiguous samples in the target domain. Unlike existing techniques, we explore a transport-based strategy that disperses target domain sample representations through learnable displacements. The objectives of such a spread would be to ease the crowding around classification boundaries (minimize class confusion [9], but novelly), *and* to align the distributions of the source and target domains [2, 13, 16, 23, 24, 29]. Additionally, there seems to be some scope left to combine useful aspects of the various categories of approaches to UDA. These observations summarize the motivation behind the proposed ATT framework for UDA. Figure 2 shows a visual description for our approach. Different components of this framework are described in each subsection below.

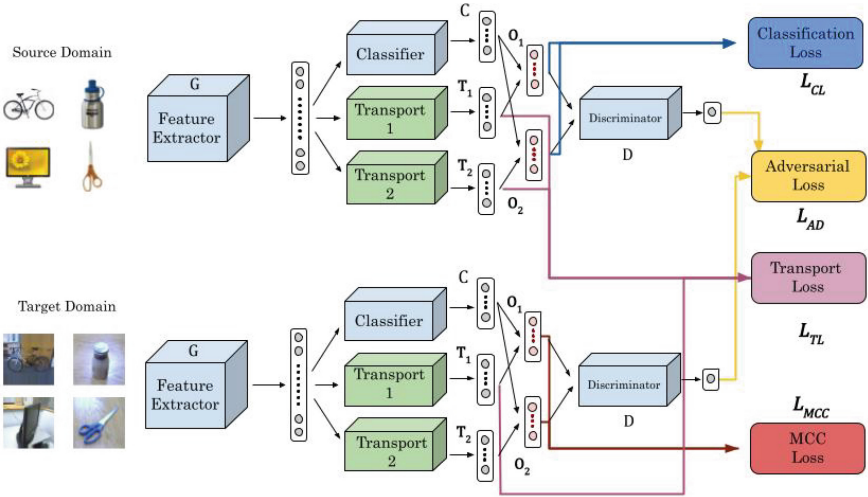


Fig. 2. The proposed ATT framework, which is shared by the source and the target domain uses Transport Layers T_1 and T_2 compute the transport vectors that move classifier outputs to their final locations.

3.1 Objective

The general problem of adversarial domain adaptation originated by ADDA [25] for classification can be formulated as follows:

$$\max_D \mathcal{L}_{adv}(G, C, D) + \mathcal{L}_c(G, C) + \mathcal{L}_e(G), \quad (2)$$

$$\min_{G, C} \mathcal{L}_{adv}(G, C, D) + \mathcal{L}_c(G, C) + \mathcal{L}_e(G), \quad (3)$$

where G is the feature extractor, C is the classifier, and D is the domain discriminator, \mathcal{L}_{adv} is the adversarial loss, and \mathcal{L}_e is the external loss as given in [25]. While a classifier head and loss may drive the labeled source samples from different classes far in the classifier (logit) space, without other objectives, such as \mathcal{L}_{adv} , the samples of the target domain may fall close to the decision boundaries. We seek to reduce the number of samples falling very close to the decision boundaries for both domains. To this end, we propose moving all the data points by a controlled amount, so that on the whole, these confusing samples become less ambiguous to classify. The amount and the direction in which the samples are moved must be learnable. For this, we propose to add two fully connected (FC) layers whose outputs we call *transport* terms in the classification (logit) space. These terms represent vectors by which we move a particular sample's representation. That is, within the general formulation of adversarial domain adaptation [25], we propose to replace the external loss by a transport loss using two transport terms \mathbf{T}_1 and \mathbf{T}_2 . We also average the two transported classifier outputs to incorporate sufficient generality. This gives rise to a more specific objective:

$$\begin{aligned} \max_D \mathcal{L}_{adv}(G, C, D, \mathbf{T}_1, \mathbf{T}_2) + \mathcal{L}_c(G, C, \mathbf{T}_1, \mathbf{T}_2) \\ + \mathcal{L}_{TL}(G, \mathbf{T}_1, \mathbf{T}_2), \end{aligned} \quad (4)$$

$$\begin{aligned} \min_{G, C, \mathbf{T}_1, \mathbf{T}_2} \mathcal{L}_{adv}(G, C, D, \mathbf{T}_1, \mathbf{T}_2) + \mathcal{L}_c(G, C, \mathbf{T}_1, \mathbf{T}_2) \\ + \mathcal{L}_{TL}(G, \mathbf{T}_1, \mathbf{T}_2), \end{aligned} \quad (5)$$

where \mathbf{T}_1 and \mathbf{T}_2 are trainable transport vectors, \mathcal{L}_{TL} is the proposed transport loss. We may not include one of the two transport terms in our experiments when there are fewer classes. The objective and the formulation of the transport are explained in more detail next.

3.2 Transport Loss

As shown in Fig. 1, when we consider samples \mathbf{x} and \mathbf{y} from two different classes, it is reasonable to assume that they are somewhat separated in the vanilla classifier space had there been no transport terms. Next, we consider a sample each from the source and the target \mathbf{x} and \mathbf{x}' , and let $C(G(\mathbf{x}))$ (abbreviated as $C(\mathbf{x})$ etc. in the figure) and $C(G(\mathbf{x}'))$ be their representations in the classifier (logit,

i.e. pre-softmax) space. We aim to move them individually by each of the two transports \mathbf{T}_1 and \mathbf{T}_2 such that the similarity between $\mathbf{T}_i(G(\mathbf{x}'))$ and $\mathbf{T}_i(G(\mathbf{x}))$ is minimized for $i \in \{1, 2\}$ and that between $\mathbf{T}_1(G(\mathbf{x}))$ and $\mathbf{T}_2(G(\mathbf{x}))$ is maximized. In the Euclidean classifier space, we employ the cosine similarity to measure similarity between two vectors. Given θ , the angle between the transport vectors, the cosine similarity is calculated as:

$$\cos(\theta) = \frac{\langle \mathbf{T}_i(G(\mathbf{x})), \mathbf{T}_i(G(\mathbf{x}')) \rangle}{|\mathbf{T}_i(G(\mathbf{x}))| |\mathbf{T}_i(G(\mathbf{x}'))|} \quad (6)$$

The above expression can be formulated into a loss function. Increasing the angles as suggested above can increase the region in which classification boundary between \mathbf{x} and \mathbf{x}' can lie, if indeed these two belong to different classes. On the other hand, if these two samples belong to the same class, their drift apart due to the transport will be compensated for by the much further drift of labeled samples from different classes. This also increases the generality of the classifier in capturing a class.

Let us formulate the transport loss. Let B denote the batch-size and c denote the number of classes. Let $A = \{a_{ij}\}$ denote the outputs of the first transport vector and $B = \{b_{ij}\}$ denote the outputs of the second transport vector. Then, an estimate of the transport yield \mathbf{Y} using the transport outputs \mathbf{T}_1 and \mathbf{T}_2 is:

$$\mathbf{Y} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,c} \\ a_{2,1} & \cdots & a_{2,c} \\ \vdots & \ddots & \vdots \\ a_{B,1} & \cdots & a_{B,c} \end{bmatrix} \begin{bmatrix} b_{1,1} & \cdots & b_{1,c} \\ b_{2,1} & \cdots & b_{2,c} \\ \vdots & \ddots & \vdots \\ b_{B,1} & \cdots & b_{B,c} \end{bmatrix}^T, \quad (7)$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_B \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_B \end{bmatrix}^T. \quad (8)$$

The resulting quantity would be a $B \times B$ matrix consisting of dot products. An additional normalizing technique is to scale the matrix entities as cosine similarities. Doing so would equalize the importance of the relationships between all the samples in the batch. In our experiments, this was seen to be favourable for accuracy in some cases but unfavourable in others.

The $B \times B$ matrix Y is an example of a bi-linear form. The most general formulation of the loss would be:

$$\mathbf{Y} = \mathbf{T}_1 \mathbf{M} \mathbf{T}_2^T, \quad (9)$$

where $\mathbf{M} = \mathbf{C}\mathbf{A}$ is a matrix which includes class information, if the samples belong to the source, and an estimation of the classes (psuedo-labels) if the samples belong to the target. The matrix \mathbf{A} is a one-hot encoding diagonal

matrix that has entries $\mathbf{A}_{ij} = 1$ if samples i, j have the same psuedolabels and $\mathbf{A}_{ij} = 0$, else. The transport loss \mathcal{L}_{TL} is then defined as:

$$\mathcal{L}_{TL} = |\Sigma(\mathbf{Y}) - Tr(\mathbf{Y})|, \quad (10)$$

where $\Sigma(\mathbf{Y})$ denotes the sum of all entries of matrix \mathbf{Y} and $Tr(\mathbf{Y})$ denotes the trace of \mathbf{Y} .

3.3 Adversarial DA with Transport Terms

Adversarial domain adaptation addresses the unsupervised domain adaptation problem by introducing an adversarial objective to the training process. Specifically, the model is trained to simultaneously minimize the classification loss on the source dataset, and to maximize the accuracy on the target dataset. A separate domain classifier is trained to distinguish between the source and target domains. The overall objective should also minimize the thus formulated adversarial transfer loss with respect to the parameters of the discriminator.

Further, the separate domain classifier, i.e. the discriminator, is trained using the following adversarial loss [5, 16]:

$$\begin{aligned} \mathcal{L}_{adv} = & -\frac{1}{N_s} \sum_{i=1}^{N_s} \log(D(G_*(x_i^s))) \\ & -\frac{1}{N_t} \sum_{j=1}^{N_t} \log(1 - D(G_*(x_j^t))), \end{aligned} \quad (11)$$

where we propose using an interpolation of the two displacements of each sample as follows:

$$G_* = \lambda \cdot (C + \mathbf{T}_1)(G) + (1 - \lambda) \cdot (C + \mathbf{T}_2)(G) \quad (12)$$

In Fig. 1, we depict this convex interpolation of \mathbf{T}_1 and \mathbf{T}_2 as \mathbf{T} . We can also rephrase the equation above as:

$$G_* = \lambda \cdot \mathbf{O}_1(G) + (1 - \lambda) \cdot \mathbf{O}_2(G), \quad (13)$$

where, G is the feature extractor, C is the classifier, \mathbf{T}_1 and \mathbf{T}_2 are the transport layers and D is the discriminator, and $\mathbf{O}_i = C + \mathbf{T}_i$ for $i \in \{1, 2\}$. The overall objectives are:

$$\max_D \mathcal{L}_{adv}, \quad (14)$$

$$\min_{G_*} \mathcal{L}_c + \mathcal{L}_{adv}, \quad (15)$$

which are done in a two-step optimization process that can be regarded as a mini-max optimization problem.

3.4 Minimum Class Confusion

The minimum class confusion loss \mathcal{L}_{MCC} [9] seeks to minimize confusion terms between classes j and j' , such that $j \neq j'$ where the indices are exhaustive over the set of classes. On the target domain, the class confusion term between two classes j and j' is given by:

$$C_{jj'} = \hat{\mathbf{y}}_{\cdot j}^T \hat{\mathbf{y}}_{\cdot j'}^T$$

A much more nuanced and meaningful formulation of the class confusion would be:

$$C_{jj'} = \hat{\mathbf{y}}_{\cdot j}^T \mathbf{W} \hat{\mathbf{y}}_{\cdot j'}^T, \quad (16)$$

where the matrix \mathbf{W} is a diagonal matrix [9]. The diagonal terms W_{ii} given as the softmax outputs of the entropies in classifying a sample i . $\hat{\mathbf{y}}_{ij}$ is given as:

$$\hat{\mathbf{y}}_{ij} = \frac{\exp(Z_{ij}/T)}{\sum_{j'=1}^c \exp(Z_{ij'}/T)}, \quad (17)$$

where c is the number of classes, T is the temperature coefficient and Z_{ij} is the logistic output of the classifier layer for the class j and the sample i .

After normalizing the class confusion terms, the final MCC Loss function is given as:

$$\mathcal{L}_{MCC} = \frac{1}{c} \sum_{j=1}^c \sum_{j' \neq j}^c |C_{jj'}|, \quad (18)$$

which is the sum of all the non-diagonal elements of the class confusion matrix. The diagonal terms represent the ‘‘certainty’’ in the classifier, while the non-diagonal terms represent the ‘‘uncertainty’’ in classification. The MCC loss can be added in conjunction with other domain adaptation methods [9]. This is not a metric learning loss that optimizes distances between samples in the classification space, but rather pushes samples that are unlike away from each other.

3.5 Overall Loss Function

The overall formulation of the total loss function will be:

$$\mathcal{L}_{TL} = |\Sigma(\mathbf{T}_1 \mathbf{T}_2^T) - Tr(\mathbf{T}_1 \mathbf{T}_2^T)|, \quad (19)$$

$$\mathcal{L}_{total} = \mathcal{L}_c + \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{TL}, \quad (20)$$

where λ_1 and λ_2 are non-negative hyperparameters. Transporting the sample representations in the classification space will be expected to reduce class confusion of the classifier. However, this may cause crowding near some of the decision boundaries in the shifted region. If \mathbf{M} is a matrix that sufficiently captures class confusions in this context, we may reduce the class confusion terms either

directly in the bilinear form $\mathbf{Y} = \mathbf{T}_1 \mathbf{M} \mathbf{T}_2^T$ so that the transport loss is modified as follows:

$$\mathcal{L}_{TL} = |\Sigma(\mathbf{T}_1 \mathbf{M} \mathbf{T}_2^T) - \text{Tr}(\mathbf{T}_1 \mathbf{M} \mathbf{T}_2^T)|, \quad (21)$$

or separately as a term in the total loss, to accelerate convergence, as follows:

$$\mathcal{L}_{total} = \mathcal{L}_c + \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{TL} + \mathcal{L}_{MCC}. \quad (22)$$

The calculation of the adversarial loss via the discriminator is done as a combination of the soft-max outputs derived from $\mathbf{O}_1 = C + \mathbf{T}_1$ and $\mathbf{O}_2 = C + \mathbf{T}_2$. The final classification task is done as:

$$\lambda \cdot \mathbf{O}_1 + (1 - \lambda) \cdot \mathbf{O}_2. \quad (23)$$

where $\lambda \in (0, 1)$. Using a linear combination makes classification less ambiguous after transporting.

4 Experiments and Results

We conducted experiments to benchmark ATT on three datasets and ablations studies to understand the contribution of its proposed components. The datasets used for benchmarking ATT were Office-Home [28], Office-31 [20], and FHIST [22].

The **Office-Home** dataset is a challenging benchmark dataset, consisting of 15,500 images across 65 classes shared by four extremely distinct domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw). All twelve transfer tasks were evaluated using this dataset [28].

The **Office-31** dataset is a widely used benchmark for visual domain adaptation, consisting of 4,110 images across 31 classes from three distinct domains: Amazon (A), Webcam (W), and DSLR (D). All six transfer tasks are evaluated using this dataset [20].

The **FHIST** dataset was originally curated for few-shot classification of near-domain target samples [22], where the source domain is CRC-TP [8] data-set and (near-domain) target is NCT-CRC-HE-100K (NCT) [10]. It consists of colorectal cancer histology images from two different domains, with 6 classes: Benign, Muscle, Stroma, Inflammatory, Debris and Tumor. For each class, there are close to 20,000 patches in the CRC-TP domain, and around 10,000 patches in the NCT domain. We rephrase this problem in terms of unsupervised domain adaptation, where we do not have access to the target labels. We explore two domain adaptation tasks of adapting the model from CRC-TP to NCT and the inverse problem. For this dataset, we exclude \mathbf{T}_2 from the training loop to favour accuracy.

We adopt the standard protocol for unsupervised domain adaptation (UDA) where all labeled source samples and unlabeled target samples are utilized for training. To report our results for each transfer task, we use center-crop images from the target domain and report the classification performance. Our experiments are conducted in PyTorch using the ResNet-50 architecture pre-trained on ImageNet.

All experiments were done on an NVIDIA RTX 3090 GPU with learning rate = 0.001 and all graphs were plotted on TensorBoard. Exploratory t-SNE plots were also generated for the source and target samples [27]. The batch size was kept 16 throughout, where 32 samples- 16 from the source and 16 from the target were used to estimate the loss in the training loop.

The weight values λ_2 used for OfficeHome and Office31 datasets are 0.0002 and 0.0016, respectively, based on the number of classes.

4.1 Benchmarking Results

As can be seen in Tables 1, 2, and 3, ATT with MCC outperforms the previous methods (including SHOT [14], SDAT [18], f-DAL [1], and GVB [3]) on all three datasets with comfortable margins.

Table 1. Accuracy (%) on the Office-Home dataset [28] with 12 different UDA tasks and their average, where all methods were fine-tuned ResNet50 [6] pre-trained on ImageNet [19]. The algorithm is adapted successfully onto an end-to-end network that uses a transformer encoder used in place of a CNN, to illustrate the versatility of the transport loss.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet-50 [6]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [5]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN [16]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
MDD [12]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
GVB-GD [3]	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
SHOT [14]	56.9	78.1	81.0	67.9	78.4	78.1	67.0	54.6	81.8	73.4	58.1	84.5	71.6
SDAT [18]	58.2	77.1	82.2	66.3	<u>77.6</u>	76.8	63.3	57.0	82.2	74.9	64.7	86.0	72.2
ATT	57.8	77.2	82.8	71.4	77.4	77.7	69.7	56.5	83.9	77.9	61	85.8	73.3
ViT [4]	66.2	84.3	86.6	77.9	83.3	84.3	75.9	62.7	88.7	80.1	66.2	88.6	78.7
CDTrans [30]	68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	90.6	80.5
ATT(ViT)	75.6	89.6	89.1	82.7	87.6	88.3	81.8	72.9	90.0	84.3	73.7	90.7	83.6

4.2 Effect of MCC and Transport Losses

The transport loss is sensitive to small increments after backpropagation. Therefore, the weight of the transport loss must be tuned for every dataset as a hyperparameter. An even higher weight in the total loss might not drive the transport loss to 0, since the updates in the parameters of \mathbf{T}_1 and \mathbf{T}_2 will overshoot the optimal values. In the formulation of the total loss given in Eq. 20 a reasonable conjecture is that the weight of the transport loss should be inversely proportional to the number of classes in the dataset. This was evident from our

Table 2. Accuracy (%) on the Office-31 dataset [20] with 6 different UDA tasks and their average, where all methods are fine-tuned ResNet50 [6] pre-trained on ImageNet [19].

Method	A \rightarrow D	A \rightarrow W	D \rightarrow W	W \rightarrow D	D \rightarrow A	W \rightarrow A	Avg
ResNet-50 [6]	68.9	68.4	96.7	99.3	62.5	60.7	76.1
DANN [5]	79.7	82.0	96.9	99.1	68.2	67.4	82.2
CDAN [16]	92.9	94.1	98.6	100.0	71.0	69.3	87.7
MDD [12]	93.5	94.5	98.4	100.0	74.6	72.2	88.9
GVB-GD [3]	95.0	94.8	98.7	100.0	73.4	73.7	89.3
SHOT [14]	93.1	90.9	98.8	99.9	74.5	74.8	88.7
f-DAL [1]	94.8	93.4	99.0	100.0	73.6	74.6	89.2
ATT	95.0	95.7	98.6	100.0	74.7	75.3	89.9

Table 3. Accuracy (%) on the FHIST dataset [22] with 2 different UDA tasks and their average, where all methods are fine-tuned ResNet50 [6] pre-trained on ImageNet [19].

Method	CRC \rightarrow NCT	NCT \rightarrow CRC	Avg
ResNet-50 [6]	40.7	32.9	36.8
DANN [5]	73.5	66.6	70.0
CDAN [16]	66.2	61.4	63.8
GVB-GD [3]	73.9	<u>66.7</u>	70.3
ATT	81.6	67.9	74.7

experiments and also by an analysis of the loss term. We also set $\lambda = 0.5$, $\lambda_1 = 1$ and $\mathbf{C} = \mathbf{I}$ (identity matrix) throughout without including the class information intrinsic to the bi-linear form. Instead, we use class information during training by using \mathcal{L}_{MCC} to accelerate the class confusion minimization.

The best results from previous domain adaptation methods are compared with our approach and the comparison is listed in Table 1.

\mathcal{L}_{MCC} directly addresses class confusion of the classifier by minimizing the probabilities of inter-class confusions. However, a careful study of our loss function on Office-Home [28] revealed that our loss addressed class confusion too, by minimizing \mathcal{L}_{MCC} , even when \mathcal{L}_{MCC} was not included in the training algorithm.

However, the converse was not true; Minimizing \mathcal{L}_{MCC} did not minimize the \mathcal{L}_{TL} , when \mathcal{L}_{TL} was not included in the training loop. We conclude that transporting the samples in the classification space in a nuanced way enables the classifier to make more confident predictions. The very idea of shifting the samples in the classification space seemed to greatly enhance accuracy, and tuning the shifting vectors \mathbf{T}_1 and \mathbf{T}_2 was but an icing on the cake, which improved the accuracy even further.

We also suspect that in certain cases, the class confusions are minimized by \mathcal{L}_{MCC} and \mathcal{L}_{TL} synergistically, so that adding the minimum class confusion loss

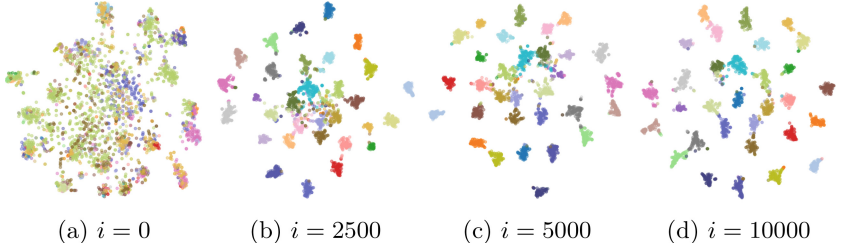


Fig. 3. Two-dimensional representation (using t-SNE [27]) of *target* samples after iterations $i = 0, 2500, 5000$ and $10,000$ of the domain adaptation task D to A on Office-31 shows that the classes initially overlap, but as the training progresses their samples are transported such that they form distinct clusters.

to the total loss caused further improvement in certain domain adaptation tasks. The class confusions that arise due to two samples from different classes being accidentally transported to nearby regions might be the cause of this.

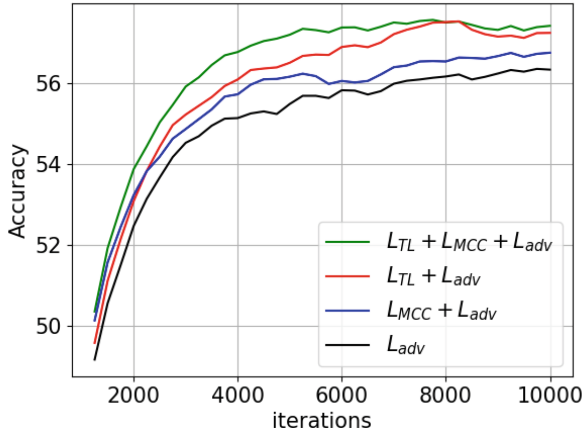


Fig. 4. Evolution of accuracy on OfficeHome [28] Ar to Cl domain adaptation task shows that improvements over using the adversarial transfer loss \mathcal{L}_{adv} [5, 16] can be made by including the minimum class confusion loss \mathcal{L}_{MCC} [9], while further and independent improvements are possible using the proposed transport loss \mathcal{L}_{TL} .

The transport loss \mathcal{L}_{TL} is sensitive to back-propagating updates, and hence to λ_2 . The sensitivity is reduced by replacing \mathcal{L}_{TL} by $\mathcal{L}_{TL_{Cos}}$ which normalizes every entry in the matrix \mathbf{Y} by the product of the norms. Empirically, the best adaptation algorithm was seen to be the loss given by Eq. 19. When the loss was modified to Eq. 21 it was observed that the convergence was slower than when the class confusion loss was included separately in the total loss. This could be possibly attributed to the back-propagation updates of \mathbf{M} being very slow, owing

to the camouflaged nature of \mathbf{M} in $\mathbf{T}_1\mathbf{M}\mathbf{T}_2^T$. Nevertheless, it is interesting to study an optimal choice of \mathbf{M} .

In our ablation studies, we observe that ATT, when combined with MCC [9] gives the best result on most of the domain adaptation tasks. The evolution of the target classifier accuracy on the domain adaptation from Ar to Cl is plotted in Fig. 4. We observe that by shifting the samples in the classification (logit) space itself produces a significantly improved accuracy compared to SOTA methods, as shown in Table 1, and tuning the transport vectors increases the accuracy. Furthermore, using the MCC loss in synergy with the transport loss favoured accuracy even more. When using just the transport loss as a means of reducing class confusion, we observe that it does a better job than the MCC loss.

4.3 Evolution of Target Domain Distribution

We visualize the movements of 2D nonlinear representation of the *target* domain feature vectors in the classification (logit) space using t-SNE (t-distributed Stochastic Neighbor Embedding) [27] after 0, 2500, 5000, and 10000 training iterations for the D to A task in Office-31. The extent of overlapping between distributions visibly reduces with increase in the number of iterations, as seen in Fig. 3.

5 Conclusions and Future Directions

We sail in previously unexplored seas of transporting samples to assist class confusion minimization. For unsupervised domain adaptation, we propose transporting samples to different regions in the classification (logit) space to allow easier joint classification of samples from source and target domains. Our results show that it is a reasonable approach to minimizing confusions in classifying the target samples in the context of unsupervised domain adaptation, especially when combined with other mechanisms, such as minimizing class confusion.

There is scope of choosing the optimal value of the confusion matrix embedded in the transport loss, by using class information of the source domain samples and an estimate of the label for the target domain samples (pseudo-labels). Doing so would gauge the effective spread of the clusters and lead to more robust classification. Moving two samples of the same class similarly might reduce the class confusion further. Given the multi-modal distribution of the source samples in the classification space, and the multi-modal prior of the target samples, we may approach the problem by introducing an angular shift term, along with linear shifting terms.

References

1. Acuna, D., Zhang, G., Law, M., Fidler, S.: f-domain-adversarial learning: theory and algorithms. In: Proceedings of the 38th International Conference on Machine Learning, pp. 66–75 (2021)

2. Chen, C., et al.: HOMM: higher-order moment matching for unsupervised domain adaptation. CoRR, abs/1912.11976 (2019)
3. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
4. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
5. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(59), 1–35 (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2015)
7. Hoffman, J., et al.: Cycada: cycle-consistent adversarial domain adaptation. In: International Conference on Machine Learning, abs/1711.03213 (2017)
8. Javed, S., Mahmood, A., Werghi, N., Benes, K., Rajpoot, N.: Multiplex cellular communities in multi-gigapixel colorectal cancer histology images for tissue phenotyping. IEEE Trans. Image Process. **29**, 9204–9219 (2020)
9. Jin, Y., Wang, X., Long, M., Wang, J.: Minimum class confusion for versatile domain adaptation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 464–480. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_28
10. Kather, J.N., Halama, N., Marx, A.: 100,000 histological images of human colorectal cancer and healthy tissue. In: NIPS (2018)
11. Le, T., Nguyen, K., Phung, D.Q.: Theoretical perspective of deep domain adaptation. CoRR, abs/1811.06199 (2018)
12. Li, J., Erpeng, C., Ding, Z., Lei, Z., Ke, L., Tao, S.: Maximum density divergence for domain adaptation. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11), 3918–3930 (2020)
13. Li, S., Liu, C., Lin, Q., Xie, B., Ding, Z., Huang, G., Tang, J.: Domain conditioned adaptation network. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, pp. 11386–11393 (2020)
14. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of the 37th International Conference on Machine Learning, **119**, 6028–6039 (2020)
15. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on Machine Learning, **37**, 97–105 (2015)
16. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
17. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Networks (2010)
18. Rangwani, H., Aithal, S.K., Mishra, M., Jain, A., Babu, R.V.: A closer look at smoothness in domain adversarial training. In: Proceedings of the 39th International Conference on Machine Learning (2022)
19. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015). <https://doi.org/10.1007/s11263-015-0816-y>
20. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16

21. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)
22. Shakeri, F., et al.: A benchmark for few-shot classification of histological images, Fhist (2022)
23. Sun, B., Saenko, K.: Deep CORAL: correlation alignment for deep domain adaptation. CoRR, abs/1607.01719 (2016)
24. Tang, H., Chen, K., Jia, K.: Unsupervised domain adaptation via structurally regularized deep clustering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
25. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
26. Vahadane, A., et al.: Structure-preserving color normalization and sparse stain separation for histological images. IEEE Trans. Med. Imaging **35**(8), 1962–1971 (2016)
27. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. (2008)
28. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation, pp. 5385–5394 (2017)
29. Wang, W., et al.: Rethinking maximum mean discrepancy for visual domain adaptation. IEEE Trans. Neural Networks Learn. Syst. **34**(1), 264–277 (2023)
30. Tongkun, X., Chen, W., Wang, P., Wang, F., Li, H., Jin, R.: Cross-domain transformer for unsupervised domain adaptation, CDTrans (2022)
31. Zhang, Y., Tang, H., Jia, K., Tan, M.: Domain-symmetric networks for adversarial domain adaptation. CoRR, abs/1904.04663 (2019)



FedSOKD-TFA: Federated Learning with Stage-Optimal Knowledge Distillation and Three-Factor Aggregation

Jianhao Liu¹, Wenjuan Gong¹(✉) , Tingbo Shi¹, Kechen Li¹, Yi Jin²,
and Jordi González³

¹ China University of Petroleum (East China), Qingdao, China
wenjuangong@upc.edu.cn

² Beijing Jiaotong University, Beijing, China

³ Autonomous University of Barcelona, Barcelona, Spain

Abstract. Federated learning is a model training method that protects user data and privacy, making it a feasible solution for multi-user collaborative training. However, due to the heterogeneity of data among clients, the optimization direction of each model is different, resulting in poor model training effects and accuracy fluctuations during training. To solve this problem, we introduce a stage-optimal strategy and propose a stage-optimal knowledge distillation method. The proposed method keeps the optimal local models and optimizes the subsequent training of the models through knowledge distillation to reduce the loss of learned knowledge. Additionally, we propose a new aggregation method that considers both static and dynamic factors. For evaluation, we conducted experiments on the CIFAR10 and CIFAR100 datasets. The proposed method significantly improved performance, achieving a maximum accuracy gain of 13.07% over the baseline model of FedPer and attaining state-of-the-art performance. The code is available at the following link: <https://github.com/FedSOKD-TFA/FedSOKD-TFA>.

Keywords: Federated Learning · Knowledge Distillation · Data Heterogeneity

1 Introduction

In recent years, data privacy and security have become a growing concern, leading to the creation of data silos among clients and resulting in data sharing issues. Traditional centralized machine learning techniques merge local datasets into one training client. Federated learning enables multiple clients to build a common, robust model without sharing data, provides a solution for multi-client collaborative training while protecting client's privacies, and has now been widely applied in biomedical imaging analysis [1], pedestrian re-identification [2], internet of things [3], etc. With advances in communication technologies, including

increased storage capacities and computing powers on edge devices, it is becoming more feasible to store data locally for training. Additionally, compared to transmitting raw data, transmitting model parameters is more efficient and cost-effective [4]. One of the commonly used assumptions in federated learning is that the data is independently and identically distributed (IID) among data silos [5], as shown in Fig. 1a. While in practical scenarios, data often appears in the form of Non-IID, which poses significant challenges to the development and application of federated learning, as illustrated in Fig. 1b. Due to data heterogeneity, each client model converges to a different position from the global model, causing the local models to deviate from the ideal optimization point. Simultaneously, this inconsistency leads to significant fluctuations in the accuracy of local models during the training process, hindering model convergence.

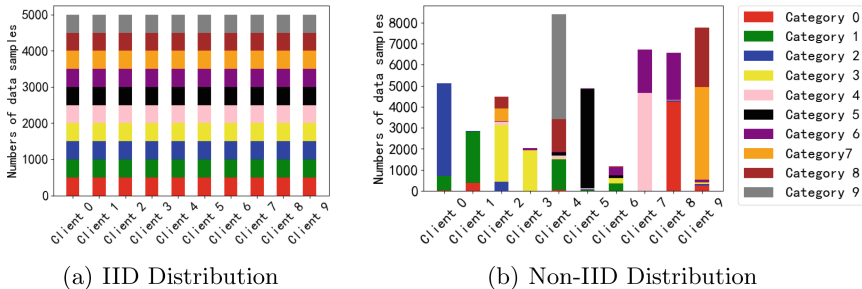


Fig. 1. IID and Non-IID Data Distributed across $K(= 10)$ Categories over $N(= 10)$ Clients.

Most of the existing federated learning algorithms solve the problem of Non-IID data distributions by modifying the local training processes and global aggregations. Strategies concentrating on local training can be mainly divided into three categories. The first constrains the training process of local clients. Related works, such as FedProx [6], SCAFFOLD [7], and MOON [8], use the global model to constrain the local training process of clients, and the drift of the model is alleviated by constraints. The second strategy alleviates the data heterogeneity through data augmentation. Related studies, such as FedMix [9], FedM-UNE [10], and FedCG [11], generate data with global information for clients to help local model training through data synthesis. The third category improves the training process through knowledge distillation. Related methods, such as FedFTG [12], FedDistill [13], DaFKD [14], and FedX [15], enhance the model performance through knowledge distillation in different directions such as global to local, local to global, clients to clients, etc. The aforementioned methods employ diverse strategies to maximize the acquisition of global knowledge by local clients, thereby enhancing model generalization. However, these approaches often overlook the critical issue of local knowledge forgetting. Due to the inherent heterogeneity of data across clients, the process of assimilating global knowledge inevitably leads to the unintended loss of locally valuable

information. To address this challenge, our proposed methodology shifts its focus towards preserving valuable local knowledge and mitigating knowledge loss. The proposed approach is rooted in the recognition that while global knowledge is indeed critical for overall model performance, the unique characteristics of local datasets often encapsulate invaluable information that should not be sacrificed in the pursuit of generalization.

On the server side, several factors were introduced to influence the process of aggregation of local models. For example, FedAvg [16] focused the amount of data, FedDisco [17] only considered the amount of data and its distribution, pFedSim [18] concentrated on the model and pFedGraph [19] assigns weights to each client by building a collaboration graph. While these factors mitigate the problem of Non-IID data distribution, the above approach considers a somewhat homogenous set of factors. This study proposes incorporating dynamic factors during model aggregation.

From the perspective of individual clients in a federated learning, the process of acquiring global knowledge is complicated by data heterogeneity. Local models may inevitably forget their homogeneous local knowledge while assimilating heterogeneous knowledge from other models. This phenomenon is known as catastrophic forgetting in the context of federated learning. Meanwhile, the overall objective of federated training is to develop a feature extractor with enhanced generalization capabilities across all participating clients. However, some current methodologies fall short in their approach to obtain this global feature extractor, as they do not comprehensively address all the nuances and challenges involved in the process.

In this work, we propose an effective client knowledge distillation method by keeping copies of the currently known optimal local models and performing knowledge distillation on these models during training on clients, and introduce a novel three-factor global aggregation strategy to optimize operations on the server side. These two modules respectively enhance model training and model aggregation within the current federated learning framework. The integration of these two approaches yields a powerful collaborative effect in federated learning. This combination effectively addresses two critical challenges: mitigating catastrophic forgetting in local models and enabling clients to acquire a global feature extractor with enhanced generalization capabilities. The contributions of this work include the followings:

1. A novel method of knowledge distillation is proposed to prevent local knowledge degradation using a stage-optimal strategy.
2. A more comprehensive model aggregation method, called three-factor aggregation, is proposed.
3. The proposed method significantly improved model performance, with a margin of 13.07% over the baseline method, FedPer [20], and outperformed the compared state-of-the-art methods, including FedAvg [16], FedProx [6], FedPer [20], pFedMe [21], and FedBN [22], achieving a SOTA performance.

2 Stage-Optimal and Three-Factor Aggregated Federated Learning

Due to the knowledge loss of local client's data and the heterogeneity of data among clients, locally learned features are prone to overfitting and have weak generalization abilities. This is a common challenge in federated learning. One of the typical solutions to this problem is Personalized Federated Learning (PFL), which aims to improve the degree of personalization of local models and the generalization ability of global models. Specifically, PFL distinguishes itself from traditional FL in its unique approach to model updates and aggregation. In each training round of PFL, clients exclusively upload the parameters of their feature extraction networks for aggregation, while retaining their local classifiers. In this study, we utilize a typical PFL system as the learning scheme. The system consists of N clients and a server. Each client $i \in \{1, 2, \dots, N\}$ holds a separate dataset $\mathcal{D}_i = \left\{ \left(x_i^j, y_i^j \right), x \in \mathcal{X}, y \in \mathcal{Y} \right\}_{j=1}^{|\mathcal{D}_i|}$, where \mathcal{X} denotes the input feature space and \mathcal{Y} corresponds to ground truth labels from K categories. And the data distribution is formulated as $P_i(x_i, y_i)$, $M_i = |\mathcal{D}_i|$, $M = \sum_{i=1}^N M_i$, where P_i is the data distribution of client i , M_i represents the number of data held by the client i , and M is the sum of data volumes across all clients.

The set of all networks are denoted as $W = \{w_1, w_2, \dots, w_i, \dots, w_N\}$. Each local client holds the decoupling model $w_i = \{\theta_i, \phi_i\}$ with a feature extractor f_θ^i and a classifier g_ϕ^i , where $f_\theta^i : \mathcal{X} \rightarrow R^d$ is a learnable network parameterized by θ_i , $g_\phi^i : R^d \rightarrow \mathcal{Y}$ is a learnable network parameterized by ϕ_i , θ_g denotes the parameters of the feature extractor after global aggregation, and d is the dimension of the feature input. Given x as input, the extracted feature vector can be denoted as $z = f_\theta^i(x)$, and the prediction of the model can be obtained through the classifier $g_\phi^i(z)$.

The proposed Federated Learning with stage-optimal knowledge distillation and three-factor aggregation (FedSOKD-TFA) approach consists of three modules, a PFL base framework, a stage-optimal knowledge distillation module, and a three-factor aggregation module. The stage-optimal knowledge module preserves local information through knowledge distillation between the stage-optimal model and the local model. And the three-factor aggregation module incorporates both static and dynamic factors for model aggregation. Additionally, we add a plug-and-play model shrinkage module on the server side. This improves the generalization ability of the model and further enhance the performance of the proposed approach. An overview of the approach is illustrated in Fig. 2.

2.1 Stage-Optimal Knowledge Distillation Strategy

The standard federated learning [16] framework initializes local models using parameters from the global model before each round of training, updating clients

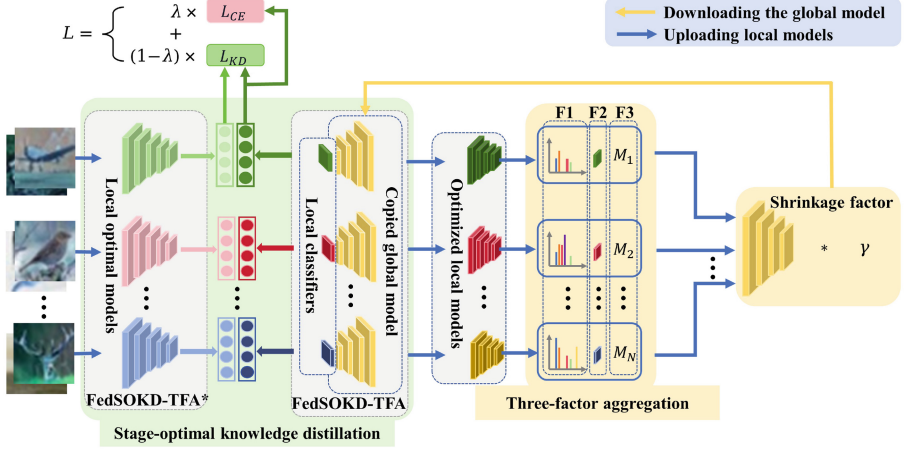


Fig. 2. Overview of the Proposed Stage-optimal and Three-factor Aggregated Federated Learning Method

with the latest global knowledge at the current stage, without emphasis on local clients. After global model aggregation and local model initialization in each round, the global model in this round may not be suitable for updating local models that were not selected and might lead to a decrease in model performance. Currently, most solutions ignore this situation, resulting in the loss of some beneficial cues for local models after global initialization and causing accuracy fluctuation during training. The proposed stage-optimal knowledge distillation strategy takes into consideration the performance of local clients and through a teacher-student methodology, the local model has access to the current global knowledge as well as reinforcing the local knowledge.

In the proposed stage-optimal knowledge distillation strategy, called SOKD, we consider keeping the existing stage-optimal model locally and transferring the knowledge of the stage-optimal model to the trained model through knowledge distillation, so that it acquires the global knowledge without losing the local optimum.

Specifically, SOKD keeps a stage-optimal model with parameters $\bar{w}_i = (\bar{\theta}_i, \bar{\phi}_i)$ for each local client and updates the teacher model accordingly. Take client i as an example, the local optimal parameter set \bar{w}_i are initialized as the optimized local parameter sets w_i in the first round. Subsequently, if the optimized model with parameters (θ_g, ϕ_i) in this round perform better than the current optimal model with parameters $(\theta_i, \bar{\phi}_i)$, this implies that the model for this round is better suited to the local data distribution than the existing optimal model, and $(\bar{\theta}_i, \bar{\phi}_i)$ is updated as (θ_g, ϕ_i) ; otherwise, the original local stage optimal model is retained. The updated local optimal model with the parameter set $\bar{w}_i = (\bar{\theta}_i, \bar{\phi}_i)$ is then employed as a teacher model. The local model is updated with the optimized parameters $w_i = (\theta_g, \phi_i)$ and is defined as the student model.

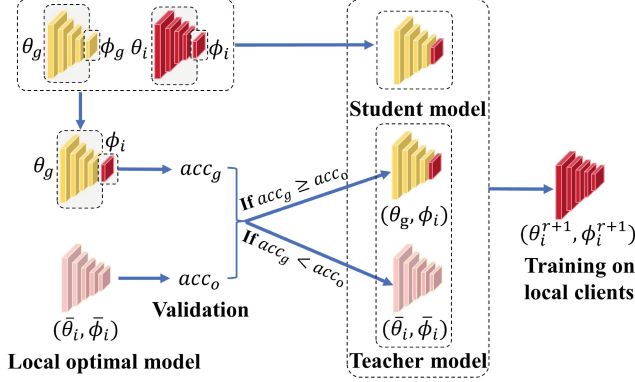


Fig. 3. The Procedure of the Proposed FedSOKD Strategy

Figure 3 illustrates the procedure. In the figure, acc_g and acc_o denote the prediction accuracy of the locally optimized model with the parameter set (θ_g, ϕ_i) and that of the stage-optimal model with the parameter set $(\bar{\theta}_i, \bar{\phi}_i)$, respectively.

Local training loss L_{local} consists of two parts, the cross-entropy loss L_{CE} between the predictions of the student model and the ground truth label, and the Kullback Leibler (KL) divergence L_{KD} between the predictions of the teacher model and the student model.

$$L_{local} = \lambda L_{CE} + (1 - \lambda) L_{KD}, \quad (1)$$

$$L_{CE}(g(f(x)), y) = -\frac{1}{M_i} \sum_{j=1}^{M_i} g(f(x_j)) \log y_j, \quad (2)$$

$$\begin{aligned} L_{KD}(g_\phi(z), g_{\bar{\phi}}(z)) &= L_{KD}(g_\phi(f_\theta(x)), g_{\bar{\phi}}(f_{\bar{\theta}}(x))) \\ &= \frac{1}{M_i} \sum_{j=1}^{M_i} g_{\bar{\phi}}(f_{\bar{\theta}}(x)) \log \frac{\text{Softmax}(g_{\bar{\phi}}(f_{\bar{\theta}}(x)/T))}{\text{Softmax}(g_\phi(f_\theta(x)/T))}, \end{aligned} \quad (3)$$

where $\lambda \in [0, 1]$ is a hyperparameter, characterizing the contributions of the two losses.

2.2 Three-Factor Aggregation Approach

In federated learning, it is necessary to determine the gap between the local model and the global model dynamically captured by the similarity of the model, and determine the weight of local model aggregation more accurately. Previous studies introduced aggregation methods using static factors, such as data distributions [17]. Aggregation weights are pre-determined by static factors during the

whole training procedure. This study proposes to combine static factors of data properties, including amount and distributions, and dynamic factors of models similarities, and propose a three-factor aggregation approach, named TFA.

The weight η_i denoting data volume for client i is calculated as the following.

$$\eta_i = \frac{M_i}{M}, \quad (4)$$

where M_i denotes the volume of data at local client i , and M represents the total volume of data across all clients.

Data distributions across categories are computed to represent data heterogeneity among clients and server. The advantage of using data distributions across categories is that it can be obtained without compromising privacy. For client i , the variance of data distributions s_i is computed as the following.

$$s_i = \sqrt{\sum_{k=1}^K (P_{i,k} - P_k)^2}, \quad (5)$$

where $P_{i,k}$ and P_k represent the local and global distributions for category k , respectively.

Previous study [18] showed that classifiers could represent model-to-model differences. Based on the observation, this study proposes to measure model similarity using cosine similarity between classifiers. The distance μ_i between the globally aggregated classifiers and the classifier of client i is measured as the following.

$$\mu_i = \max \left(\frac{\phi_g \cdot \phi_i}{\|\phi_g\|_2 \|\phi_i\|_2}, 0 \right), \quad (6)$$

where ϕ_i denotes the parameters of the classifier on client i in the current training round.

Once these weights are computed for each client, they are stored in vectors, $\eta = [\eta_1, \eta_2, \dots, \eta_n]$, $s = [s_1, s_2, \dots, s_n]$ and $\mu = [\mu_1, \mu_2, \dots, \mu_n]$, for data volume, client's category distribution difference, and model difference, respectively. It is important to note that if client i is not selected for participation in a given round of federated learning, its corresponding value in each vector is set to 0. Based on these vectors, TFA applies a linear rectifier, and calculates the global weight θ_g as a weighted sum of parameters from all selected clients.

$$p = \text{Relu}(\eta - a \times s - b \times \mu + c), \quad (7)$$

$$\theta_g = \sum_{i=1}^{N \times \zeta} \frac{p_i}{p} \theta_i, \quad (8)$$

where a , b , and c are hyperparameters, p is the aggregated weight after applying the linear rectifier, and ζ is active ratio. After applying the Relu (Rectified Linear Unit) activation function, we obtain a rectified vector, which effectively introduces non-linearity while preserving the dimensionality of the original vector. Hyperparameters a and b serve as critical diversity weights in TFA module,

playing a important role in balancing the various components of our algorithm. To further refine our approach, we introduce parameter c as a y measure. The primary purpose of c is to mitigate potential over-correction that may arise from suboptimal configurations of a and b .

2.3 FedSOKD-TFA Algorithm

To provide a concise overview of the proposed method, we present the algorithm in pseudocode format below. The algorithm is divided into two parts: one for the client-side operations and the other for the server-side operations. The TFA method, included in the pseudocode, is described in detail in Sect. 2.2.

Algorithm 1. FedSOKD-TFA

Require: Local optical model $(\bar{\theta}_i, \bar{\phi}_i)$, global feature extractor θ_g , local epoch E , rounds R , dataset $\{D_{train}^i, D_{test}^i\}$, active ratio ζ , number of clients N

- 1: **Initialization** : local models $w = [w_1, w_2, \dots, w_N]$ and global model w_g
- 2: **Client** :
- 3: **for** $r = 0$ to $R - 1$ **do**
- 4: Selected clients $w_{selected} = \text{RandomSelect}(w, N \times \zeta)$
- 5: **if** $r == 0$ **then**
- 6: $\bar{w} = [\bar{w}_1, \bar{w}_2, \dots, \bar{w}_N] = [w_1, w_2, \dots, w_N]$
- 7: **else**
- 8: Download global feature extractor θ_g .
- 9: Attach θ_g and ϕ_i to w_i .
- 10: **for** w_{temp} in $w_{selected}$ **do**
- 11: Compute $\text{acc}((\bar{\theta}_i, \bar{\phi}_i), (\theta_g, \phi_i^{r-1}), D_{test}^i) \rightarrow acc_o, acc_g$
- 12: **if** $acc_g > acc_c$ **then**
- 13: Update local optical $(\bar{\theta}_i, \bar{\phi}_i) \rightarrow (\theta_g, \phi_i)$
- 14: **else**
- 15: Continue
- 16: **end if**
- 17: Get Teacher model $\bar{w}_i = (\bar{\theta}_i, \bar{\phi}_i)$ and student model $w_i = (\theta_i^r, \phi_i^r)$
- 18: $\text{eval}(\bar{w}_i)$
- 19: **for** $e = 0$ to $E - 1$ **do**
- 20: $w_i^e \leftarrow$ local training for τ steps of SGD
- 21: **end for**
- 22: Send $w_{tmp}^{trained}$ to server
- 23: **end for**
- 24: **end if**
- 25: **end for**
- 26: **Server** :
- 27: Get selected models $w^{trained}$, client data volume M , client data distribution P_i .
- 28: Compute each weights using Eq. 7
- 29: Aggregate using TFA
- 30: Distribute the aggregated feature extractor θ_g

The overall structure of the algorithm is as above. In the first round, the global model, local models, and local optimal models are initialized. In subsequent rounds, the local models are initialized at the beginning of each round to determine the local optimal models for that round. After completing the training, the models are uploaded to the server and aggregated using the TFA method.

2.4 FedSOKD-TFA: Convergence Analysis

We begin by introducing the following assumptions:

Assumption 1. *Lipschitz Smoothness.* Gradients of client i 's local complete heterogeneous model w_i are L_1 - Lipschitzsmooth,

$$\begin{aligned} \|\nabla \mathcal{L}_i^{t_1}(w_i^{t_1}; x, y) - \nabla \mathcal{L}_i^{t_2}(w_i^{t_2}; x, y)\| &\leq L_1 \|w_i^{t_1} - w_i^{t_2}\|, \\ \forall t_1, t_2 > 0, i \in \{0, 1, \dots, N-1\}, (x, y) \in D_i \end{aligned} \quad (9)$$

The above formulation can be further derived as:

$$\mathcal{L}_i^{t_1} - \mathcal{L}_i^{t_2} \leq \langle \nabla \mathcal{L}_i^{t_2}, (w_i^{t_1} - w_i^{t_2}) \rangle + \frac{L_1}{2} \|w_i^{t_1} - w_i^{t_2}\|_2^2 \quad (10)$$

Assumption 2. *Unbiased Gradient and Bounded Variance.* Client i 's random gradient $g_t^{w,i} = \nabla \mathcal{L}_i^t(w_i^t, \mathcal{B}_i^t)$, (\mathcal{B} is a batch of local data) is unbiased,

$$\mathbb{E}_{\mathcal{B}_i^t \subseteq D_i} [g_{w,i}^t] = \nabla \mathcal{L}_i^t(w_i^t) \quad (11)$$

and the variance of random gradient $g_t^{w,i}$ is bounded by:

$$\mathbb{E}_{\mathcal{B}_i^t \subseteq D_i} [\|\nabla \mathcal{L}_i^t(w_i^t; \mathcal{B}_i^t) - \nabla \mathcal{L}_i^t(w_i^t)\|_2^2] \leq \sigma^2 \quad (12)$$

Assumption 3. *Bounded Parameter Variation.* The parameter variations of the homogeneous small feature extractor θ_i^t and θ^t before and after aggregation is bounded as

$$\|\theta^t - \theta_i^t\| \leq \delta^2 \quad (13)$$

Through Assumptions 1 and 2, we can establish the following lemma,

Lemma 1

$$\mathbb{E}[\mathcal{L}_t^{E+1}] \leq \mathcal{L}_t^{E+0} + \left(\frac{L_1 \eta^2}{2} - \eta\right) \sum_{e=1}^E \|\nabla \mathcal{L}_t^{E+e}\|_2^2 + \frac{L_1 \eta^2 \sigma^2}{2} \quad (14)$$

Leveraging Assumption 3 and Lemma 1, we derive Lemma 2.

Lemma 2

$$\mathcal{L}_{t+1}^{E+0} = \mathcal{L}_{t+1}^E + \mathcal{L}_{t+1}^{E+0} - \mathcal{L}_{t+1}^E \approx \mathcal{L}_{t+1}^E + \eta \|\theta_{t+1}^{E+0} - \theta_{t+1}^E\|_2^2 \leq \mathcal{L}_{t+1}^E + \eta \delta^2 \quad (15)$$

By synthesizing Lemma 1 and 2, we ultimately arrive at the following conclusion:

$$\eta < \frac{2(\epsilon - \delta^2)}{L_1(\epsilon + E\sigma^2)} \quad (16)$$

Given that ϵ , L_1 , δ^2 , σ^2 are all positive constants, it follows that η has well-defined solutions. Consequently, when the learning rate η satisfies the aforementioned condition, convergence is assured for any client's local complete heterogeneous model. For a comprehensive proof and detailed derivation, please refer to our code repository: <https://github.com/FedSOKD-TFA/FedSOKD-TFA/blob/main/Convergence.pdf>.

3 Experimental Results

To rigorously evaluate the efficacy of the proposed federated learning approach, comprehensive experiments were conducted on two widely-adopted and challenging visual benchmark datasets: CIFAR-10 and CIFAR-100. These datasets have emerged as standards for assessing the performance of computer vision models, particularly in the realm of image classification tasks. The Cifar10 dataset includes 10 categories, each category consisting of 6000 RGB images with resolutions of 32×32 . Among them, 5000 are used as training and 1000 as testing. The Cifar100 dataset contains 100 classes, each class consisting of 600 RGB images with resolutions of 32×32 . Among them, 500 are used as training and 100 as testing. By benchmarking the proposed federated learning approach on these widely-adopted and challenging datasets, the study aims to provide a comprehensive and objective assessment of its performance.

To investigate the real-world applicability of the proposed federated learning method, the experimental data was generated using a Non-IID setting. Specifically, the data were partitioned across clients according to a Dirichlet distribution such that the categories were unevenly distributed. The Dirichlet distribution is controlled by a concentration parameter β . A smaller value of β corresponds to a higher degree of Non-IID data distribution among the clients. The experiments evaluated the proposed approach under two Non-IID scenarios, with β set to 0.1 and 0.5, respectively, representing significant and moderate deviations from the IID setting. The shrinkage factor γ , which controls the degree of model regularisation, was set to 0.95. The hyperparameter λ , governing the relative importance of the distillation loss term in the overall objective function, was assigned a value of 0.8. Furthermore, the distillation temperature T , which modulates the softening of the logits during knowledge distillation, was set to 7. Finally, the active ratio ζ , determining the fraction of clients participating in each round of federated learning, was fixed to 0.5.

The backbone network used in the experiments is a five-layered CNN. The federated learning environment encompasses a total of 10 clients, with 50% of these clients actively participating in each round of the training process. The optimization procedure involves 200 iterative rounds of global communication, wherein each round incorporates 5 epochs of localized training on the respective

Table 1. Prediction Accuracy of the Proposed Method Compared with the State-of-the-art Methods on Two Standard Evaluation Datasets. Numbers in brackets are the differences of prediction accuracy between the proposed method and the best performing method.

Method	Accuracy \uparrow Predicted with Different Settings of β			
	Cifar10		Cifar100	
	$\beta = 0.1$	$\beta = 0.5$	$\beta = 0.1$	$\beta = 0.5$
FedAvg [16]	58.34	64.54	24.1	27.27
FedProx[6]	60.55	64.38	24.9	27.38
FedPer [20]	90.38	78.84	46.06	33.52
FedPer [20]+WeightShrinking [23]	90.05	79.67	49.14	34.89
pFedMe [21]	89.61	74.59	41.7	29.12
FedBN [22]	87.83	74.7	44.07	35.2
FedSOKD-TFA	91.03 (32.69 \blacktriangle)	82.71 (18.17 \blacktriangle)	57.61 (33.51 \blacktriangle)	44.1 (16.83 \blacktriangle)
FedSOKD-TFA*	91.18 (32.84 \blacktriangle)	83.59 (19.05 \blacktriangle)	59.13 (35.03 \blacktriangle)	45.52 (18.25 \blacktriangle)

\uparrow The upward-pointing arrow indicates that the algorithm performed better as the prediction accuracy increases.

\blacktriangle The black triangle pointing upwards indicates that the proposed method performed better.

client devices. For local training, the optimiser is set to SGD, the batch size to 64, and the learning rate to 0.01.

3.1 Comparison with the State-of-the-Art Methods

To validate the efficiency of the proposed approach, we conducted a comprehensive comparative evaluation against the state-of-the-art methodologies in the field of federated learning. The methods included in this analysis are FedAvg [16], FedProx [6], FedPer [20], pFedMe [21], FedBN [22], and WeightShrinking is a regularisation method in [23]. The experimental results are listed in Table 1. In this table, we report two distinct accuracy measures: the accuracy of the global model, denoted as FedSOKD-TFA, and the accuracy of the best-performing local model, represented as FedSOKD-TFA*. These models allow for a detailed understanding of the proposed method’s performance at both the global and local levels, enabling a holistic assessment of its capabilities.

A thorough examination of the results presented in Table 1, reveals a compelling observation, the proposed FedSOKD-TFA approach exhibited a clear superiority over all the compared state-of-the-art methods, consistently outperforming them across two distinct datasets. This empirical evidence substantiates the effectiveness and robustness of the FedSOKD-TFA technique, underscoring its potential as a leading solution in the field of federated learning. Notably, when the hyperparameter β was set to 0.1, the performance of FedSOKD-TFA* surpassed that of FedPer [20], initialized with a global model, by a remarkable margin of 13.07%. This significant performance differential highlights the substantial improvements offered by the proposed approach over existing methodologies. Furthermore, the FedSOKD-TFA method itself demonstrated a

performance improvement of up to 11.55% under the same experimental setting. This empirical evidence strongly validates the argument that simply initialising the local client using the global model from the previous round may not be the most appropriate approach, as it may hinder the optimisation process and limit overall performance.

Compared to FedAvg [16], the standard federated learning method, FedSOKD-TFA* and FedSOKD-TFA gained a boost of 35.03% and 33.51%, respectively, with the same settings as above. These results demonstrate that the locally optimal model is not only more friendly to local data, but the globally aggregated model also achieves better performance compared to other approaches.

3.2 Ablation Studies

We conducted ablation experiments and evaluated the effect of each module, as listed in Table 2. With $\beta = 0.5$ and all other parameters identical to the above settings, experimental results showed that the prediction accuracy of the FedSOKD method that uses the same aggregation methodology as FedPer [20] outperformed FedPer [20] by 3.07% on the Cifar10 dataset and by 7.56% on the Cifar100 dataset. This demonstrates that is effective by adding local copies of the optimal models, and during the training process, the local models acquire learned knowledge from the locally saved copies.

We also experimented with three different aggregation methods, weighing various factors, including the amount of data in FedPer [20], amount of data and data distribution in FedDisco [17], and multiple factors in FedTFA. Experimental results showed that FedTFA outperformed the other two aggregation methods. This suggests that TFA’s idea of considering both dynamic and static factors in the training process is feasible. Additionally, the combination of FedSOKD and FedTFA achieved the highest performance among all configurations. These two modules can be used together to achieve better performance, or they can be split and applied to different federated learning frameworks. For instance, when combined with the SOKD strategy, the performance of the FedDisco [17] and FedPer [20] methods enhanced by it, respectively.

3.3 Experiments on Parameter Settings

To evaluate the effectiveness of the weight shrinking module, we tested its boosting effect on the FedPer and FedSOKD methods. The experimental results on the Cifar10 and Cifar100 datasets are presented in Table 3. Compared to the FedPer method, the improvement achieved by weight shrinking on the FedSOKD method was 2.19%, indicating that weight shrinking complements the proposed method more effectively.

We conducted experiments to evaluate the performance of the proposed method by varying model parameters. The loss function comprises two components. The weight of each component is controlled by the parameter λ , while the temperature of the distillation loss during training is governed by the parameter T . The experiments were carried out using the FedSOKD-TFA method on

Table 2. Ablation Studies on the SOKD and TFA Modules.

Method	Dataset	Parameters	Accuracy \uparrow
FedPer [20]	Cifar10	–	78.84
	Cifar100	–	33.52
FedDisco [17]	Cifar10	$a = 0.5$ $b = 0.1$	79.03
	Cifar100	$a = 0.5$ $b = 0.1$	33.2
FedTFA	Cifar10	$a = 0.6$ $b = 0.3$ $c = 0.3$	79.28
	Cifar100	$a = 0.6$ $b = 0.3$ $c = 0.3$	33.5
FedSOKD-Per [20]	Cifar10	$T = 7$ $\lambda = 0.8$	81.91
	Cifar100	$T = 7$ $\lambda = 0.8$	41.08
FedSOKD-Disco [17]	Cifar10	$a = 0.5$ $b = 0.1$	81.88
	Cifar100	$a = 0.5$ $b = 0.1$	41.25
FedSOKD-TFA	Cifar10	$T = 7$ $\lambda = 0.8$	82.23
	Cifar100	$T = 7$ $\lambda = 0.8$	41.49

\uparrow The arrow indicates that the algorithm performed better as the prediction accuracy increases.

Table 3. Impact of Weight Shrinking. The numbers in brackets represent the differences in prediction accuracy between the method with weight shrinking and the method without.

Method	Accuracy \uparrow	
	Cifar10	Cifar100
FedPer	78.84	33.52
FedPer+WeightShrinking	79.67 (0.83 \blacktriangle)	34.89 (1.37 \blacktriangle)
FedSOKD	81.02	40.12
FedSOKD+WeightShrinking	82.66 (1.64 \blacktriangle)	43.68 (3.56 \blacktriangle)

\uparrow The upward-pointing arrow indicates that the algorithm performed better as the prediction accuracy increases.

\blacktriangle The black triangle pointing upwards indicates that the proposed method performed better.

the Cifar10 dataset, with β set to 0.5. To demonstrate the impact of parameter variations, this study performed the experiments by fixing one parameter and altering the value of the other. Table 4 presents the parameters used in our experiments and their corresponding prediction accuracy.

As evident from the Table 4, the performance of FedSOKD-TFA is more sensitive to variations in the parameter λ , whereas the distillation temperature T does not significantly impact the model’s accuracy. The value of λ can be fine-tuned to accommodate different datasets or various data distribution strategies employed during the federated learning process.

To assess the sensitivity of three hyperparameters in the TFA module, we conducted a series of experiments. In these experiments, we systematically varied one parameter while keeping the other two fixed. The experiment’s results as

Table 4. Prediction Accuracy of the FedSOKD-TFA Method with Different λ and T on the Cifar10 Dataset.

Prediction Accuracy with Fixed λ or T			
$\lambda = 0.8$		$T = 5$	
Value of T	Accuracy \uparrow	Value of λ	Accuracy \uparrow
5	83.66	$\lambda = 0.4$	81.3
10	83.93	$\lambda = 0.5$	82.6
15	83.79	$\lambda = 0.6$	83.14
20	83.87	$\lambda = 0.7$	83.65
25	83.63	$\lambda = 0.8$	83.66
30	83.76	$\lambda = 0.9$	83.86

\uparrow The upward-pointing arrow indicates that the algorithm performed better as the prediction accuracy increases.

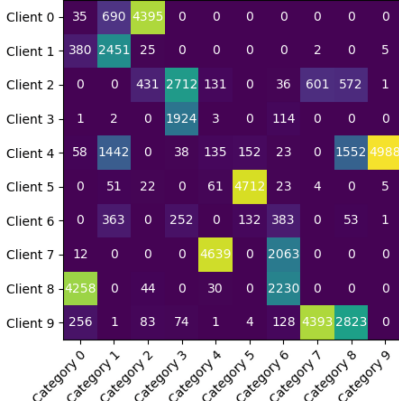
Table 5. Prediction Accuracy of the FedSOKD-TFA Method with Different a , b and c on the Cifar10 Dataset.

Prediction Accuracy with Fixed a or b or c				
Parameters			Algorithm	
a	b	c	FedSOKD-TFA	FedSOKD-TFA*
0.1	0.3	0.3	82.95	83.99
0.3	0.3	0.3	83.42	84.08
0.5	0.3	0.3	83.1	84.04
0.7	0.3	0.3	82.82	83.96
0.9	0.3	0.3	82.71	83.94
0.3	0.1	0.3	82.81	84.00
0.3	0.3	0.3	83.42	84.08
0.3	0.5	0.3	82.9	84.1
0.3	0.7	0.3	82.72	83.94
0.3	0.9	0.3	82.11	83.58
0.3	0.3	0.1	82.6	83.91
0.3	0.3	0.3	83.42	84.08
0.3	0.3	0.5	83.07	83.9
0.3	0.3	0.7	82.86	83.96
0.3	0.3	0.9	82.72	83.85

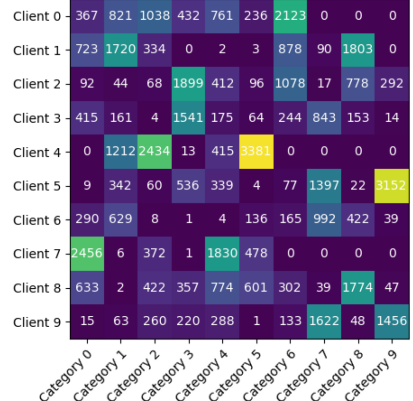
showed Table 5 Our findings indicate that the accuracy of the optimal model (FedSOKD-TFA*) is relatively insensitive to parameter changes. However, when concatenating the global feature extractor with the local classifier (FedSOKD-TFA), we observed that different parameter settings significantly impact the accuracy.

3.4 Data Distribution and Detailed Experimental Results

To provide a better visualization of the data distribution across clients for different values of β , we present heat maps of the Cifar10 dataset for various β values with 10 clients in Fig. 4.

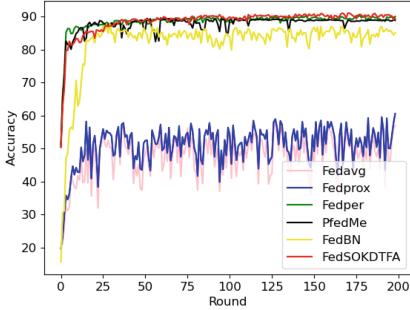


(a) Data distribution with $\beta = 0.1$

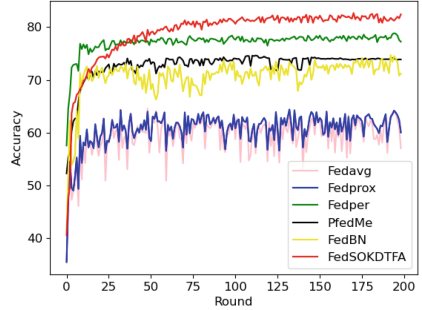


(b) Data distribution with $\beta = 0.5$

Fig. 4. Data Distributions with Two Settings ($\beta = 0.1$ and $\beta = 0.5$) on the Cifar10 Dataset.



(a) Training accuracy plot with $\beta = 0.1$



(b) Training accuracy plot with $\beta = 0.5$

Fig. 5. Training Accuracy of Two Non-IID Data Distributions on the Cifar10 Dataset.

Additionally, we display the training accuracy when β is set to 0.1 and 0.5 in Fig. 5. We observe that as the value of β decreases, the heterogeneity among clients becomes more pronounced, leading to a greater disparity in data distribution, with the data being concentrated in fewer classes. Through the analysis of

Fig. 4 and the experiments depicted in Fig. 5, we conclude that the proposed Personalized Federated Learning (PFL) approach achieves better performance when β is smaller, indicating a higher degree of data heterogeneity across clients. Conversely, standard federated learning methods like FedAvg perform better when β is larger, implying a more homogeneous data distribution.

An examination of the accuracy curves in Fig. 5, reveals that the proposed FedSOKD-TFA method exhibits smaller fluctuations in accuracy across both settings. This observation indicates that the proposed method mitigates the accuracy regression phenomenon to a certain degree by preventing the loss of local knowledge. Concurrently, the method achieves superior accuracy compared to all the other methods depicted in the graph, as evidenced by the higher accuracy values attained throughout the training process.

4 Discussion

The combination of SOKD and TFA represents a balanced approach to mitigate catastrophic forgetting of local knowledge and obtain a more generalizable feature extraction network. On one hand, SOKD enables local models to acquire global knowledge while minimizing the loss of local knowledge. On the other hand, TFA helps obtain a feature extraction network with stronger generalization capabilities relative to all participating clients. However, due to the incorporation of knowledge distillation, SOKD module requires two forward passes through the same data, which increases the consumption of computational resources. The computational overhead introduced by the TFA method is negligible compared to that of SOKD. In the future work, we will explore the possibility of implementing SOKD for selected rounds, which aims to investigate the potential for both mitigating local knowledge forgetting and reducing computational overhead.

Currently, the practical applications of federated learning have progressed beyond theoretical concepts. Previous studies have already initiated the implementation of federated learning in real-world scenarios, spanning diverse fields such as cybersecurity [24], healthcare [25], and natural language processing [26]. However, our research to date has primarily focused on theoretical image classification tasks. In the upcoming phase of our investigation, we aim to expand our exploration of this methodology to real-world datasets and evaluate its potential to address tangible, practical challenges.

5 Conclusion

In this paper, we proposed to address the limitations of the original framework that used a global model to initialize the local model. SOKD is a client-side approach that stores the knowledge learned by the local model by retaining a copy of the optimal model at current stage locally. Then, the local knowledge is transferred to the current initialized model by means of knowledge distillation during local training. During the training process, the trained model carries

the global knowledge on one hand and acquires the local knowledge on the other hand, which makes the knowledge richer and the training model is thus further optimized. Meanwhile, the TFA method combines dynamic and static factors, providing a better option for global model aggregation. The experiment proved that the proposed methods obtained a significant performance improvement compared to other methods.

Acknowledgements. Wenjuan Gong acknowledges the support by the Natural Science Foundation of Shandong Province under Grant ZR2023MF041. Jordi González acknowledges the support of the Spanish Ministry of Economy and Competitiveness (MINECO) and the European Regional Development Fund (ERDF) under Project No. PID2020-120611RBI00/AEI/10.13039/501100011033.

References

1. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. *J. Healthc. Inf. Res.* **5**, 1–19 (2021)
2. Zhuang, W., et al.: Performance optimization of federated person re-identification via benchmark analysis. In: *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 955–963 (2020)
3. Li, Z., et al.: Data heterogeneity-robust federated learning via group client selection in industrial IoT. *IEEE Internet Things J.* **9**(18), 17844–17857 (2022)
4. Su, X., Zhou, Y., Cui, L., Liu, J.: On model transmission strategies in federated learning with lossy communications. *IEEE Trans. Parallel Distrib. Syst.* **34**(4), 1173–1185 (2023)
5. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. *arXiv preprint [arXiv:1806.00582](https://arxiv.org/abs/1806.00582)* (2018)
6. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2**, 429–450 (2020)
7. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: stochastic controlled averaging for federated learning. In: *International Conference on Machine Learning*, pp. 5132–5143. PMLR (2020)
8. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722 (2021)
9. Yoon, T., Shin, S., Hwang, S.J., Yang, E.: FedMix: approximation of mixup under mean augmented federated learning. *arXiv preprint [arXiv:2107.00233](https://arxiv.org/abs/2107.00233)* (2021)
10. Zhang, H., Hou, Q., Wu, T., Cheng, S., Liu, J.: Data augmentation based federated learning. *IEEE Internet Things J.*, 1 (2023). <https://doi.org/10.1109/JIOT.2023.3303889>
11. Wu, Y., et al.: FedCG: leverage conditional GAN for protecting privacy and maintaining competitive performance in federated learning, pp. 2309–2315 (2022). <https://doi.org/10.24963/ijcai.2022/321>
12. Zhang, L., Shen, L., Ding, L., Tao, D., Duan, L.Y.: Fine-tuning global model via data-free knowledge distillation for non-IID federated learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10174–10183 (2022)

13. Jiang, D., Shan, C., Zhang, Z.: Federated learning algorithm based on knowledge distillation. In: 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), pp. 163–167. IEEE (2020)
14. Wang, H., Li, Y., Xu, W., Li, R., Zhan, Y., Zeng, Z.: DaFKD: domain-aware federated knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20412–20421 (2023)
15. Han, S., et al.: FedX: unsupervised federated learning with cross knowledge distillation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13690, pp. 691–707. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20056-4_40
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
17. Ye, R., Xu, M., Wang, J., Xu, C., Chen, S., Wang, Y.: FedDisco: federated learning with discrepancy-aware collaboration. In: Proceedings of the 40th International Conference on Machine Learning, ICML 2023. JMLR (2023)
18. Tan, J., Zhou, Y., Liu, G., Wang, J.H., Yu, S.: pFedSim: similarity-aware model aggregation towards personalized federated learning. arXiv preprint [arXiv:2305.15706](https://arxiv.org/abs/2305.15706) (2023)
19. Ye, R., Ni, Z., Wu, F., Chen, S., Wang, Y.: Personalized federated learning with inferred collaboration graphs. In: International Conference on Machine Learning, pp. 39801–39817. PMLR (2023)
20. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint [arXiv:1912.00818](https://arxiv.org/abs/1912.00818) (2019)
21. Dinh, C.T., Tran, N., Nguyen, J.: Personalized federated learning with Moreau envelopes. Adv. Neural Inf. Process. Syst. **33**, 21394–21405 (2020)
22. Li, X., Jiang, M., Zhang, X., Kamp, M., Dou, Q.: FedBN: Federated learning on non-IID features via local batch normalization. In: International Conference on Learning Representations (2021), <https://openreview.net/forum?id=6YEQU0QICG>
23. Li, Z., Lin, T., Shang, X., Wu, C.: Revisiting weighted aggregation in federated learning with neural networks. In: Proceedings of the 40th International Conference on Machine Learning, ICML 2023. JMLR (2023)
24. Liu, Y., et al.: FedVision: an online visual object detection platform powered by federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 13172–13179 (2020)
25. Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A.: FedDG: federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1013–1023 (2021)
26. Lin, B.Y., et al.: FedNLP: benchmarking federated learning methods for natural language processing tasks. arXiv preprint [arXiv:2104.08815](https://arxiv.org/abs/2104.08815) (2021)



DualViT: A Hierarchical Vision Transformer for Broad and Fine Class Embeddings

Ankita Chatterjee¹(✉), Sandip Dutta¹, Jayanta Mukhopadhyay¹,
and Partha Pratim Das^{1,2}

¹ Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India

ankita.chatterjee25@kgpian.iitkgp.ac.in, {jay,ppd}@cse.iitkgp.ac.in

² Department of Computer Science, Ashoka University, Sonapat, Haryana, India

Abstract. Human brains are natural learning systems which inherently recognise image objects in a hierarchical pattern. Similar association exists among different categories of images which interact while training a deep learning model, leading to misclassification and indistinct features. As semantically related classes form clusters in the embedding space, we can list these classes under a single aggregate class, and make the learning faster by leveraging this knowledge while training a model. In this paper, we introduce a hierarchical architecture that simultaneously learns both broad-class and fine-class embeddings using transformer-based encoders. These encoders are trained alternately, and the hierarchical knowledge is encoded through tensor product between the coarse and fine class representations. Our approach has been able to achieve state-of-the-art performance on benchmark datasets with very less epochs.

Keywords: Vision transformer · Class embedding · Hierarchical learning

1 Introduction

Hierarchical learning architectures have been able to achieve significant improvements in various computer vision tasks, such as, image classification, object detection and semantic segmentation. It categorises the visual features into hierarchical structures, mimicking the recognition approach of human visual system. Through hierarchical feature extraction and representation learning, these architectures capture meaningful visual embeddings at multiple levels of abstraction, facilitating robust and distinct interpretations of visual data.

Recently, a novel hierarchical attention mechanism is proposed in [4] which uses the relationship among regions to generate captions of images. Similarly, spatial features at multiple scales for a scene are modeled in [8] using a hierarchical graph convolutional network (GCN). The reported network leverages

First and Second authors have equal contribution.

hierarchical graph representations to achieve state-of-the-art results in scene parsing. Researchers have shown keen interest in utilising hierarchical features in emerging domains such as video understanding, 3D scene reconstruction, and autonomous driving. In [2], a method to reduce the severity of mistakes is proposed by incorporating hierarchical knowledge while training a model primarily focusing on autonomous driving. In [15], authors introduce hierarchical spatiotemporal GCN to capture both spatial and temporal features in video action recognition.

Image classification tasks have previously seen hierarchical knowledge being integrated with visual features to increase the accuracy of the model. For example, [6, 29, 32] devised methods to quantify the similarity among classes to reduce associated training costs. While [5] proposed a similarity matrix method to embed hierarchy, it is mostly focused on enhancing image retrieval efficiency rather than classification. Despite advancements in the performance of classifiers, utilising inter-class relationships for image classification remains unexplored. Although recent techniques such as [3, 30] leverage hierarchical knowledge through soft label and hierarchical prompting, respectively, to enhance the performance of the model as well as learn meaningful features, these methods do not specifically learn embeddings of broad and fine classes simultaneously.

In this paper, we propose a novel architecture, *DualViT*, with two *ViT* [9] encoders dedicated to generate embeddings of broad and fine labels, respectively. They are trained using alternate learning to extract features of both coarse and fine categories of classes. The hierarchical knowledge is embedded from the broad-class encoder to the fine-class encoder using tensor product [26] computation. We utilise the semantic relationship among the classes to generate class hierarchy from *WordNet* ontology [20]. Our main objectives are as follows:

1. Generating embeddings for both broad and fine classes using alternate learning of encoders in a single architecture.
2. Modifying the loss function and introducing embedding loss to map each of the fine classes to its corresponding broad class.
3. Training fine-class encoder from scratch to achieve comparable performance with the conventional classification models using hierarchical knowledge from broad-class encoder through tensor product.

2 Related Work

The primary objective of most classification models is to optimize accuracy of prediction, without considering the semantic nature of errors. This can result in models that achieve great levels of performance, but also exhibit perplexing errors, such as incorrectly classifying two unrelated classes. Most recent works focus on understanding of the model to estimate its behaviour under adversarial attack [1]. Due to this shift in the interest of researchers, image classification methods using taxonomic structure of semantically related classes to improve the overall accuracy of the model have been extensively explored in [23, 25, 27].

However, these models aim to learn the associations present in the dataset rather than leveraging the existing hierarchical patterns present among the classes.

Authors of [28], apply Gaussian priors, based on a fixed hierarchy, on weights of neurons in a neural network. On the other hand, [22] use a tree-like graphical model during inference which is derived from class taxonomy. While [10] propose regularization based method which penalises misclassified samples with a higher cost, [14] guide their training which reduces the prediction cost and enhances the performance of the model. They define the hierarchies apriori and form cost-based matrix which is used as a guiding metric. However, Bertinetto et al. [2] highlighted the lack of using hierarchical nomenclatures in computer vision. They proposed two classification models based on soft labels derived using least common ancestor method [5] and hierarchical cross-entropy loss. Although, these methods reduced the extent of misclassification, they mostly found its application in image retrieval.

HiMulConE [31] introduces hierarchy preserving loss considering the class representations in the embedding space. It is based on the assumption that classes with same ancestors are relatively closer in the latent space rather than two different classes. A similar approach is used in [3] where the hierarchy is derived from the *WordNet* ontology using which soft labels were defined. These labels were used along with cross-entropy loss to train a classification model. Karthik et al. [11] aim at minimising the average hierarchical cost by reducing the inference time risk using predicted posteriors. TransHP [30] use hierarchical prompting to include broad-class prompts as intermediate features in fine-class encoding blocks while training.

Recently, hyperbolic spaces have become the focus in modeling hierarchical relations due to their ability to incorporate hierarchical data structures into low-dimensional spaces [12, 21, 24]. Authors of [17, 18] suggest incorporating a class hierarchy into the latent representation space. Nevertheless, both methods encode class hierarchy into the data embedding network before training. On the other hand, it has been observed that embedding the hierarchical structure while training the model enables the network and class embeddings to interact, resulting in a better trade-off between hierarchical cost and accuracy.

Although, the recent methods have seen substantial improvement in image classification task, all these work focus on improving fine-class accuracy rather than learning both broad and fine class embeddings. We propose a novel architecture that learns both the representations as well as injects the broad class information as features while training the fine-class encoder for classification.

3 Proposed Method

In this section, we propose a method to deploy hierarchical knowledge while training fine classes from scratch, meanwhile learning broad-class representations simultaneously. Our main aim is to stimulate faster convergence with high classification accuracy along with meaningful feature space.

3.1 Forming Class Hierarchies

WordNet [20] ontology forms a lexical database capturing the representations of the semantics present in a sentence. The image classes in *ImageNet* dataset are derived from the leaf nodes of a sub-tree structure present in *WordNet* hierarchy. This tree encapsulates semantic relationships between nodes. By leveraging the *WordNet* taxonomy embedded within the *ImageNet* dataset, we can effectively establish the semantic associations among its classes. The leaf nodes constitute the fine labels while the branch nodes denote aggregate classes. We consider branch nodes at a particular depth from the root as broad labels while the leaves connected to that sub-tree as fine labels associated with it.

Depth Estimation. We observe that the hierarchical tree for the *ImageNet* dataset becomes imbalanced at certain depths of the tree. For example, at depth 3 from the root, some branch nodes (broad classes) are parents to a large number of leaves (fine classes), while the others contained very few classes in their sub-tree. Therefore, it is important that we adopt a strategy to estimate the depth of the tree from where we will segregate the broad and fine classes in a manner that it is mostly balanced.

Let the maximum and the minimum depth of the tree from root to any leaf node be d_{max} and d_{min} , respectively. We expand each path with depth $< d_{max}$ by adding pseudo branch nodes between the root and the leaves. The path to leaf nodes which are already at d_{max} depth are kept unaltered. After modifying the tree, we select depth $d > d_{min}$ and consider all the nodes in that level as broad classes and the leaves of each sub-tree under that broad class as fine classes. This strategy splits all the branch nodes with large number of fine classes into smaller sub-classes, thus, mitigating the issue of class imbalance to a large extent as shown in Fig. 1. In our experiments, we have fixed the broad class depth at $d = 9$. In case of *CIFAR10* and *CIFAR100* dataset, we obtain the class hierarchy from [2] directly. Table 1 shows the number of broad classes obtained for each depth in *ImageNet* class hierarchy tree.

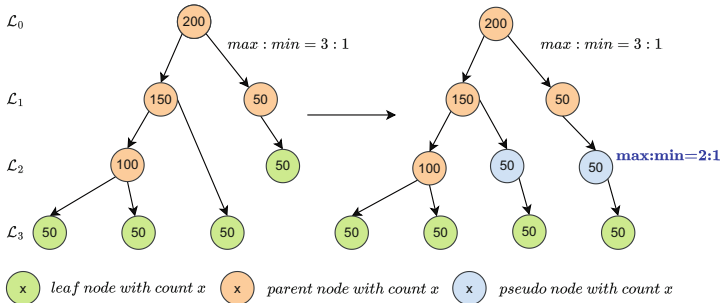


Fig. 1. Formation of the class hierarchy using pseudo nodes for depth estimation and balanced class segregation.

Table 1. Variation in the number of broad classes for different depth of hierarchy tree formed for *ImageNet* dataset.

Depth	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Broad classes	1	2	5	22	47	112	240	403	587	736	819	887	921	977

3.2 Network Architecture

We propose a transformer-based architecture, *DualViT*, that comprises of two encoders (broad-class, \mathbb{E}_b and fine-class, \mathbb{E}_f encoders), each for generating broad and fine class image embeddings, respectively. Our main idea is to leverage the broad class groupings while training the fine classes to stimulate faster convergence and comparable results with the traditional classification models. The encoder architecture is similar to the existing *ViT* [9] model. A schematic diagram of the proposed architecture is shown in Fig. 2.

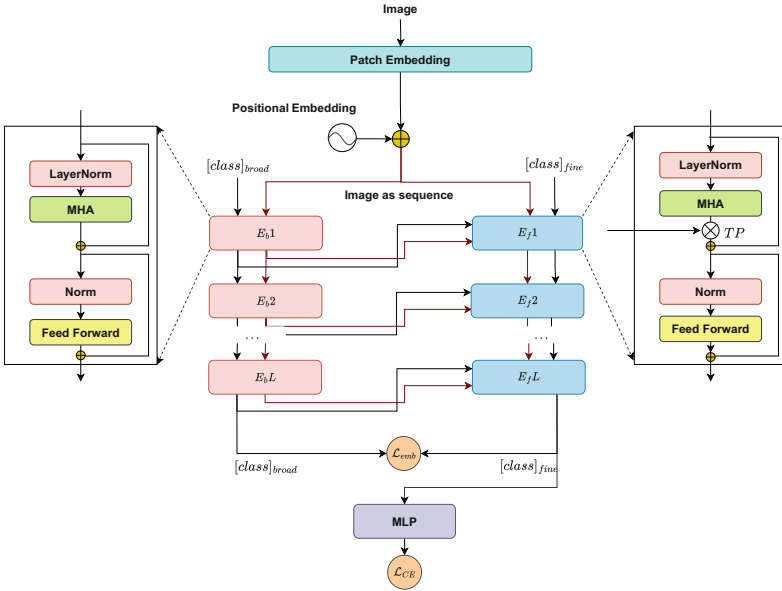


Fig. 2. Schematic representation of the *DualViT* architecture.

Let (H, W) be the image dimensions and C be the number of channels, therefore, the 2-D input can be represented as $x \in \mathbb{R}^{H \times W \times C}$. The input images are split into 2-D patches given by $x_p \in \mathbb{R}^{N \times (P^2 \times C)}$, where, (P, P) is the patch

resolution and $N = \frac{HW}{P^2}$. These patches are converted into fixed latent dimension, D , by applying (P, P) convolution filters with stride P . The number of kernels is fixed at D . Thus, we generate N vectors of dimension D which are regarded as patch embeddings. We prepend a special [class] token (κ) of dimension D to the patch embedding. To retain positional information and provide some inductive bias, we add $(N + 1) \times D$ positional embedding to the patch embedding. This tensor, denoted as z_0 , is passed to the broad-class, \mathbb{E}_b and fine-class, \mathbb{E}_f encoders.

Encoder Blocks. The encoder \mathbb{E} consists of L layers of encoder blocks T_{b_i} , $i = 1, 2, \dots, L$. Each of these blocks have sequential stacking of LayerNorm and Multi-head attention (MHA) module which is followed by another stack of LayerNorm and Position-wise feed forward layer. The multi-head attention (MHA) captures global context using attention mechanism. This forms the core of transformer-based models containing h heads which can operate in parallel. Each element in the given input sequence is transformed into “query”, “key”, and “value” vectors using learnable projections, W_i^k, W_i^q, W_i^v , respectively. Attention mechanism is applied on these vectors to generate scores for all the heads present. The scores from all the heads are concatenated and transformed using a learnable projection W^o . The position-wise feed forward layer operates on each layer independently. Thus, the network learns non-linear relationships between features within each token, complementing the context captured by self-attention. Shared weights across positions ensure efficient parameter usage while allowing for position-specific transformations. The broad-class encoder, \mathbb{E}_b follows this architecture which is similar to the conventional ViT encoder \mathbb{E} .

Tensor Product. Encoder \mathbb{E}_f is connected to \mathbb{E}_b through tensor product between the outputs of the corresponding blocks present in \mathbb{E}_b encoder. The result of the tensor product generated from each block is propagated to the next block in \mathbb{E}_f while \mathbb{E}_b propagates the output of the position-wise feed forward layer. Although, \mathbb{E}_b can be trained independently, the input to each block in \mathbb{E}_f depends on the output of the previous block of \mathbb{E}_b . In this way we are able to leverage the aggregate class information from \mathbb{E}_b while training \mathbb{E}_f . The output of the i^{th} encoder block, T_{f_i} , of \mathbb{E}_f is given by:

$$T_{f_i}[z_{i-1}] = T_{b_i}[z_{i-1}] * \psi_{i-1} \quad (1)$$

where, $T_{f_i}[z_{i-1}]$ and $T_{b_i}[z_{i-1}]$ are the outputs of the i^{th} encoder block of \mathbb{E}_f and \mathbb{E}_b , respectively, $< * >$ denotes element-wise product, ψ_{i-1} is the output of the multi-head attention (MHA) layer of T_{f_i} and z_{i-1} is the input to T_{f_i} . The [class] tokens from \mathbb{E}_f (κ_f) are sent to the classifier to generate softmax probabilities which we represent as $\kappa_m = p(\kappa_f)$. Furthermore, we utilise the [class] tokens from \mathbb{E}_b (κ_b) to compute the embedding loss.

3.3 Class Embedding

Feature representations of similar classes form clusters in the latent space which can be regarded as a group of classes denoting an aggregate class. The representation of this broad class can be estimated by computing the mean of all these similar class embeddings. Let C and B denote all the fine and broad classes, respectively, and e_{ij} denote the embedding of the i^{th} sample belonging to the j^{th} class, where, $j \in C$. Therefore, each of these class representations is given by:

$$\kappa_f^j = \frac{\sum_{i=1}^n e_{ij}}{n} \quad (2)$$

where, κ_f^j is the embedding vector of class j and n is the number of samples present for a class in a particular batch. If S_1 be a set of similar classes, then the broad class embedding for S_1 will be:

$$\kappa_b^{S_1} = \frac{\sum_{j=1}^{|S_1|} \kappa_f^j}{|S_1|} \quad (3)$$

Thus, κ_b is formed for each such similar groups of class embeddings.

3.4 Loss Function

We train \mathbb{E}_b using embedding loss computed between the fine-class embeddings, κ_f^j , $j \in S$ and its corresponding broad-class embeddings, κ_b^S , where S is a set of similar classes. The broad class sets are generated using the class hierarchies formed. The embedding loss, \mathcal{L}_{EMB} for each mini-batch K is given by:

$$\mathcal{L}_{EMB} = \sum_{i=1}^K \sum_{j=1}^K (\kappa_b^i - \kappa_f^j) \delta_{ij} \quad (4)$$

where, i and j denote the broad and fine classes, respectively, and

$$\delta_{ij} = \begin{cases} 1, & j \in i \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

The fine-class encoder, \mathbb{E}_f is trained using a combination of cross-entropy loss and embedding loss. As the loss scale varies significantly, we use logarithm normalisation to map the higher values to lower ranges. The final loss \mathcal{L} is given by:

$$\mathcal{L} = \log(\mathcal{L}_{EMB}) + \mathcal{L}_{CE} \quad (6)$$

where, \mathcal{L}_{CE} is the cross-entropy loss using fine-class labels.

3.5 Training Strategy

Our main idea is to leverage the broad level grouping of fine classes during training of fine-class encoder, \mathbb{E}_f , as well as learning the broad class embeddings by mapping these class representations to their corresponding groups. If both the encoders are backpropagated at the same time, the model fails to converge. On the other hand, if we train the two encoders separately in a consecutive manner for higher number of epochs, the loss becomes significantly large with reduced impact of hierarchical knowledge. Therefore, to induce the effects of similar class grouping, we train the encoders alternatively for 80 epochs.

In the alternating training strategy, for every odd epoch the weights of the broad-class encoder, \mathbb{E}_b , is updated while the weights of the \mathbb{E}_f is updated for every even epoch. The input is sent to the pair of encoders and the embeddings are generated for both the fine and broad classes in every epoch. If the number of epoch is odd, only the \mathbb{E}_b is trained using embedding loss computed between the fine classes and its corresponding broad class embeddings. On the other hand, if the number of epoch is even, the \mathbb{E}_f is trained using a combination of embedding loss and cross-entropy loss.

4 Experimentation

We evaluate our proposed approach on *CIFAR10* [13], *CIFAR100* [13] and *ImageNet* [7] datasets and compare it with the existing classification techniques which use hierarchical knowledge while training.

4.1 Data Set

ImageNet [7] is a widely used benchmark dataset from Large Scale Visual Recognition Challenge (ILSVRC) 2012 with 1000 fine labels. The dataset itself does not embed hierarchical relationships present among the classes. However, each of these classes are present as leaf nodes of the *WordNet* [20] ontology which is used to form broad and fine categories based on least common ancestor approach present in [5, 29]. We divide the 1.28 million images given as training samples into training and validation sets with a ratio of 80% – 20%. The 50,000 validation samples are kept aside as test set. Similar set-up is employed across all the models used for comparison.

CIFAR10 [13] dataset has 50,000 training samples for 10 classes, with each comprising of 5000 examples. The testing set has 1000 samples for each. Thus, a total of 60,000 images are present in *CIFAR10* dataset. The same number of total samples are present in *CIFAR100* [13] dataset which are equally distributed among 100 classes. Each class consists of 500 and 100 images for training and testing, respectively. The size of the images for both *CIFAR10* and *CIFAR100* is 32×32 .

4.2 Training Set-Up

The broad-class, \mathbb{E}_b , and the fine-class encoder, \mathbb{E}_f , of our proposed *DualViT* model are trained alternately for 80 epochs using stochastic gradient descent (SGD) and nesterov accelerated gradient (NAG), with a weight decay of 10^{-6} . As the \mathbb{E}_b layer weights are derived from the pre-trained *ViT* model, we keep the learning rate low at 10^{-5} , and fine-tune these layers with the embedding loss. However, the \mathbb{E}_f is trained from scratch using a combination of embedding and cross-entropy loss. Therefore, the initial learning rate is kept high at 10^{-3} , which slowly decays following the cosine annealing scheduler. The image size is fixed to 224 for *ImageNet*, while it is 32 for *CIFAR10* and *CIFAR100* datasets. The batch size is kept at 64 for all the datasets due to our constrained resources. We use a system with the following specifications, CPU: Intel(R) Xeon(R) Gold 5118 CPU @ 1.30 GHz, RAM: 251 GB, GPU: Quadro GV100 32 GB.

4.3 Evaluation Metrics

The top-k accuracy measure is a frequently employed method for evaluating classifiers. It determines whether a data point is correctly classified by examining if the actual class label is included among the top k predicted classes with the highest probabilities. Traditionally, the value of k is set to either 1 or 5. It is important to mention that this metric treats all errors equally, regardless of the proximity of the predicted class to the actual value. We compute the top-k accuracy for both the broad and fine classes present in the dataset.

4.4 Results and Discussions

We generate the top-1 and top-5 accuracy of our proposed *DualViT* model on both broad and fine classes keeping the batch size fixed to 32, 32 and 64 for *CIFAR10*, *CIFAR100* and *ImageNet* datasets, respectively. The *CIFAR10* dataset has 2 broad classes each having 4 and 6 fine classes respectively, while *CIFAR100* segregates 100 classes equally into 20 coarse categories. In case of *ImageNet*, we have been able to identify 736 broad classes using our pseudo node addition method for class hierarchy. Table 2 shows the top-1 and top-5 accuracy observed for our model. We further plot fine class embeddings to visualise how similar classes form broad groups in the embedding space using t-SNE

Table 2. Top-1 and top-5 accuracy of broad and fine classes of *CIFAR10*, *CIFAR100* and *ImageNet* datasets.

Datasets	Broad classes		Fine classes	
	top-1 (in %)	top-2/top-5 (in %)	top-1 (in %)	top-5 (in %)
CIFAR10	99.59	100.0 (top-2)	98.11	99.97
CIFAR100	98.14	99.77	90.61	98.69
ImageNet	94.93	98.63	78.85	94.02

[16, 19] plots in Fig. 3. We observe how semantically related classes have formed separate groups depicting a broad class in the embedding space.

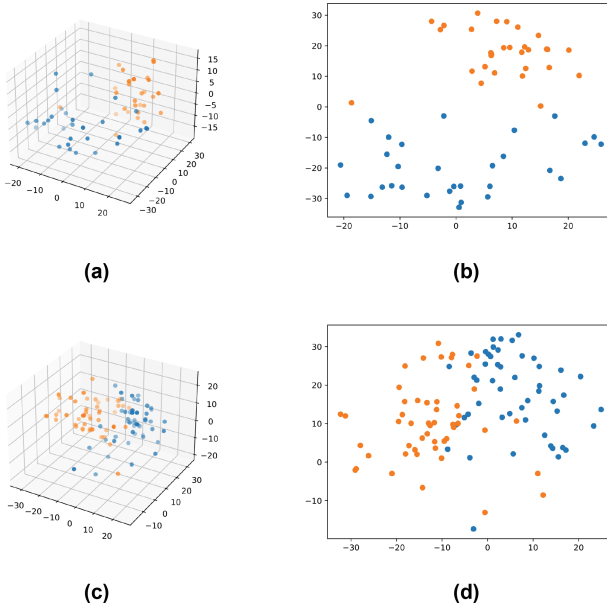


Fig. 3. t-SNE plots of fine classes with its corresponding coarse categories in the latent space using *CIFAR10* dataset. (a) & (b) denote 3D and 2D plots of randomly selected 40 sample points. (c) & (d) denote 3D and 2D plots of randomly selected 100 sample points. Color orange: “bird”, “cat”, “dog”, “deer”, “frog”, “horse” classes. Color blue: “airplane”, “automobile”, “ship”, “truck” classes. (Color figure online)

4.5 Comparative Study

Our method has been compared to the existing state-of-the-art techniques [3, 14, 30, 31] which impose hierarchical knowledge while training for fine-class classification. Authors of [14] employ a guided training of feature extractor which uses the relative arrangement of classes in the embedding space regarded as hierarchical metric. HiMulConE [31] use a novel hierarchical loss which combines contrastive loss with hierarchical penalty to preserve the relationship among the classes while training. In [3], the authors have enhanced the quality of feature space by using soft labels while training a *ViT* model. The soft labels denote the relationship among the classes derived using least common ancestor method from *WordNet* hierarchy. On the other hand, [30] uses hierarchical prompting to

generate and insert tokens of coarse classes into intermediate features. We compare our method with the existing state-of-the-art techniques in Table 3 using *CIFAR100* and *ImageNet* datasets. Our model, *DualViT* outperforms the existing methods using very less epochs.

Table 3. Comparing the top-1 and top-5 accuracy of our *DualViT* model with the state-of-the-art techniques on *CIFAR100* and *ImageNet* datasets.

Method	CIFAR100		ImageNet	
	top-1 (in %)	no. of epochs	top-1 (in %)	no. of epochs
One-hot encoding (Baseline)	84.98	300	76.21	350
Metric-Guided [14]	85.10	200	76.05	300
HiMulConE [31]	85.43	300	77.52	100
Soft-labels encoding [3]	87.95	300	74.50	80
TransHP [30]	86.85	300	78.65	300
DualViT (ours)	90.61	80	78.85	30

4.6 Study of Convergence

We observe that by leveraging the class hierarchy while training facilitates faster convergence. We are able to achieve benchmark results within very few epochs. We compare how the loss decreases with the increase in epoch in Fig. 4 for *DualViT* with Soft-labels based training [3] and TransHP [30].

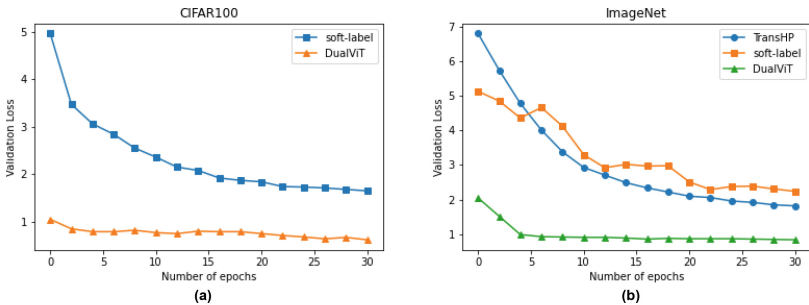


Fig. 4. Comparing the rate of convergence of *DualViT* with Soft-labels based training [3] and TransHP [30] using (a) *CIFAR100*, and (b) *ImageNet* datasets.

From Fig. 4, we observe how fast the loss decreases in case of *DualViT* compared to [3, 30]. As it starts from a very less loss margin, it converges much faster. Further, we compare the accuracy per epoch on validation set of *CIFAR100* and *ImageNet* datasets in Fig. 5.

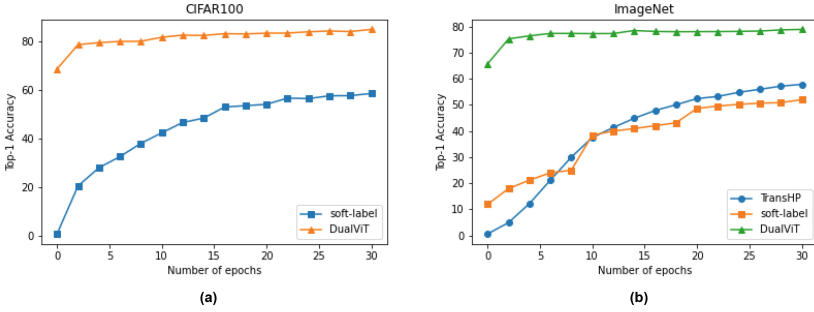


Fig. 5. Comparing the accuracy per epoch of *DualViT* with Soft-labels based training [3] and TransHP [30] on validation set of (a) *CIFAR100*, and (b) *ImageNet* datasets.

4.7 Ablation Studies

We evaluate the impact of using embedding loss along with cross-entropy loss to train the fine classes, and the significance of tensor product to connect the components of broad and fine-class encoders in our ablation studies. We perform all our ablation studies on *CIFAR10* and *CIFAR100* datasets.

Impact of Embedding Loss. The feature representations of semantically related classes are not significantly discriminative. Thus, while training a model, similar classes mostly get misclassified. They tend to form groups in the embedding space denoting a larger set of classes. Our main objective behind computing the embedding loss is to minimize the distance between the embedding of a broad class and the embedding of a fine class belonging to that broad class. The intuition behind this objective function is that as the fine class is a subset of that broad class, its representation should lie within the hypersphere formed by the broad class embeddings in the latent space. In Table 4 we observe how the inclusion of embedding loss improves the fine-class accuracy of the model.

Significance of Tensor Product. Tensor product transformer attention [26] is prominently used in the domain of natural language processing and machine translation. This attention mechanism extends the traditional transformer architecture by introducing tensor products to enhance the ability of the model to capture complex interactions between different parts of the input. By computing tensor products between the query, key, and value vectors, the model can effectively learn intricate relationships and dependencies within the input data. This approach enables the model to capture relevant features across multiple dimensions simultaneously. We learn the relationships between the broad-class and fine-class representations by computing the tensor product between the output of each block of \mathbb{E}_b and \mathbb{E}_f , and propagate the resultant value as input to the next block in \mathbb{E}_f . In Table 4 we analyse the impact of tensor product on the fine-class accuracy of the model for all the datasets.

Table 4. Analysing the impact of embedding loss and tensor product on top-1 and top-5 accuracy of *CIFAR10* and *CIFAR100* datasets.

Parameters	<i>CIFAR10</i>				<i>CIFAR100</i>			
	<i>Broad classes</i>		<i>Fine classes</i>		<i>Broad classes</i>		<i>Fine classes</i>	
	top-1 (in %)	top-2 (in %)	top-1 (in %)	top-5 (in %)	top-1 (in %)	top-5 (in %)	top-1 (in %)	top-5 (in %)
w/ embedding loss	99.59	100.0	98.11	99.97	98.14	99.77	90.61	98.69
w/o embedding loss	46.77	100.0	97.96	99.93	9.54	40.18	88.96	98.08
w/ tensor product	99.59	100.0	98.11	99.97	98.14	99.77	90.61	98.69
w/o tensor product	95.73	100.0	73.94	95.91	12.13	47.29	47.94	61.01

From Table 4, we observe that there has been significant improvement in the broad class top-1 accuracy when both embedding loss and tensor product is used. However, the fine class accuracy remains similar as the impact of cross-entropy loss is predominant while training in this case.

4.8 Hyperparameter Tuning

Hyperparameter tuning is a critical task which enhances the performance of the model if the parameters are selected correctly. If these parameters are not appropriately adjusted, our model estimates may yield subpar results. Due to the parallel processing capabilities of GPUs, researchers often lean towards training their models with larger batches. However, it has been observed that employing a large batch size can lead to suboptimal generalization. Empirically it has been found that smaller batch sizes exhibit faster training and better generalization on unseen test data. Nevertheless, extremely small batch sizes, such as 1, typically yields poor performance. Therefore, in general, the batch size is usually greater than the number of classes present in a training set based on the assumption that there should be at least one sample of each class present in a batch for training. We observe the variation in top-1 and top-5 accuracy on *CIFAR10*, *CIFAR100* and *ImageNet* datasets with different batch sizes in Table 5.

Table 5. Analysing the variation in top-1 and top-5 accuracy of *CIFAR10*, *CIFAR100* and *ImageNet* datasets on different batch sizes.

Batch Size	<i>CIFAR10</i>				<i>CIFAR100</i>				<i>ImageNet</i>			
	<i>Broad classes</i>		<i>Fine classes</i>		<i>Broad classes</i>		<i>Fine classes</i>		<i>Broad classes</i>		<i>Fine classes</i>	
	top-1 (in %)	top-2 (in %)	top-1 (in %)	top-5 (in %)	top-1 (in %)	top-5 (in %)	top-1 (in %)	top-5 (in %)	top-1 (in %)	top-5 (in %)	top-1 (in %)	top-5 (in %)
16	96.78	100.0	97.33	99.94	86.29	99.22	88.62	98.11	35.46	66.29	78.42	94.43
32	99.58	100.0	98.11	99.97	98.14	99.77	90.61	98.69	42.51	66.16	78.75	94.37
64	95.80	100.0	97.11	99.94	67.23	96.40	87.98	98.16	92.61	98.63	78.85	94.02

From Table 5, we observe that the optimal batch size for *CIFAR10* and *CIFAR100* is 32 while the accuracy for *ImageNet* keeps increasing with the increase in the batch size. Conventionally, a batch size of 4096 is used in ViT [9].

5 Conclusions

In this paper, we propose an architecture, *DualViT*, to efficiently learn broad-class and fine-class embeddings simultaneously. Moreover, the encoders interact using tensor product to leverage the hierarchical knowledge from the broad-class encoder on the fine-class encoder to train fine labels for better accuracy. This not only facilitates faster learning but also ensures meaningful feature space with classes around the true label being semantically related to each other. We have compared our method with the existing state-of-the-art techniques and observe that *DualViT* surpasses them by a significant margin.

References

1. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: a survey. *IEEE Access* **6**, 14410–14430 (2018)
2. Bertinetto, L., Mueller, R., Tertikas, K., Samangooei, S., Lord, N.A.: Making better mistakes: leveraging class hierarchies with deep networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12506–12515 (2020)
3. Chatterjee, A., Mukherjee, J., Das, P.P.: ImageNet classification using wordnet hierarchy. *IEEE Trans. Artif. Intell.* **5**, 1718–1727 (2023)
4. Chen, H., Ding, G., Lin, Z., Guo, Y., Shan, C., Han, J.: Image captioning with memorized knowledge. *Cogn. Comput.* **13**, 807–820 (2021)
5. Deng, J., Berg, A.C., Fei-Fei, L.: Hierarchical semantic indexing for large scale image retrieval. In: *CVPR 2011*, pp. 785–792. IEEE (2011)
6. Deng, J., Berg, A.C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010*. LNCS, vol. 6315, pp. 71–84. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15555-0_6
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE (2009)
8. Deng, Y., et al.: Hierarchical memory learning for fine-grained scene graph generation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *ECCV 2022*. LNCS, vol. 13687, pp. 266–283. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19812-0_16
9. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
10. Hou, L., Yu, C.P., Samaras, D.: Squared earth mover’s distance-based loss for training deep neural networks. *arXiv preprint arXiv:1611.05916* (2016)
11. Karthik, S., Prabhu, A., Dokania, P.K., Gandhi, V.: No cost likelihood manipulation at test time for making better mistakes in deep networks. In: *International Conference on Learning Representations* (2020)
12. Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., Lempitsky, V.: Hyperbolic image embeddings. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6418–6428 (2020)
13. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

14. Landrieu, L., Garnot, V.S.F.: Leveraging class hierarchies with metric-guided prototype learning. In: British Machine Vision Conference (BMVC) (2021)
15. Lee, J., Lee, M., Lee, D., Lee, S.: Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10444–10453 (2023)
16. Li, W., Cerise, J.E., Yang, Y., Han, H.: Application of t-SNE to human genetic data. *J. Bioinform. Comput. Biol.* **15**(04), 1750017 (2017)
17. Liu, S., Chen, J., Pan, L., Ngo, C.W., Chua, T.S., Jiang, Y.G.: Hyperbolic visual embedding learning for zero-shot recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9273–9281 (2020)
18. Long, T., Mettes, P., Shen, H.T., Snoek, C.G.: Searching for actions on the hyperbole. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1141–1150 (2020)
19. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008)
20. Miller, G.A.: WordNet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995)
21. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. *Adv. Neural Inf. Process. Syst.* **30** (2017)
22. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
23. Roy, D., Panda, P., Roy, K.: Tree-CNN: a hierarchical deep convolutional neural network for incremental learning. *Neural Netw.* **121**, 148–160 (2020)
24. Sala, F., De Sa, C., Gu, A., Ré, C.: Representation tradeoffs for hyperbolic embeddings. In: International Conference on Machine Learning, pp. 4460–4469. PMLR (2018)
25. Salakhutdinov, R., Tenenbaum, J.B., Torralba, A.: Learning with hierarchical-deep models. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1958–1971 (2012)
26. Schlag, I., Smolensky, P., Fernandez, R., Jojic, N., Schmidhuber, J., Gao, J.: Enhancing the transformer with explicit relational encoding for math problem solving. arXiv preprint [arXiv:1910.06611](https://arxiv.org/abs/1910.06611) (2019)
27. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. *Data Min. Knowl. Disc.* **22**, 31–72 (2011)
28. Srivastava, N., Salakhutdinov, R.R.: Discriminative transfer learning with tree-based priors. *Adv. Neural Inf. Process. Syst.* **26** (2013)
29. Verma, N., Mahajan, D., Sellamanickam, S., Nair, V.: Learning hierarchical similarity metrics. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2280–2287. IEEE (2012)
30. Wang, W., Sun, Y., Li, W., Yang, Y.: TransHP: image classification with hierarchical prompting. *Adv. Neural Inf. Process. Syst.* **36** (2024)
31. Zhang, S., Xu, R., Xiong, C., Ramaiah, C.: Use all the labels: a hierarchical multi-label contrastive learning framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16660–16669 (2022)
32. Zhao, B., Li, F., Xing, E.: Large-scale category structure aware image categorization. *Adv. Neural Inf. Process. Syst.* **24** (2011)



Establishing Interconnections of Similarity-Based Classifiers for Multi-label Learning with Missing Labels

Sambhav Jain  and Reshma Rastogi ^(✉) 

Machine Learning and Statistical Inference (MLSI) Lab, Department of Computer
Science, South Asian University, Delhi, India

reshma.khemchandani@sau.ac.in

Abstract. Similarity-based classification framework is extensively used to address the problem of multi-label learning. Through this research, we establish the connection between similarity-based classification with many popular state-of-the-art multi-label learning models. In fact, we show that the similarity-based classification framework shares homology with Support Vector Machines. Further, we explore the application of the proposed framework to deal with the problem of multilabel learning with missing labels. Our models require only the plugin estimates for sample-sample and label-label similarity, which are cohesed into a single term that leads to a parameter-free optimisation problem. Hence, The proposed model design is simple and time-efficient. Further, we have proposed two transductive models and two lazy learners which can be used as per applicability. The models shows competitive performance with other state-of-the-art models across five well-known multilabel datasets.

Keywords: Similarity based multi-label learning · Support Vector
Machines · Laplacian matrix · Multi-label learning with Missing Labels

1 Introduction

In a multi-label learning problem, a sample may belong to more than one label simultaneously, in contrast to the multi-class scenario where a sample can have only one label. Note that in multi-label setting, the labels can be related in some sense to each other, giving rise to this domain's signature property, which is label-label correlation. This condition sets multilabel learning apart from other classification paradigms.

Multi-label learning is a supervised learning scenario, but practically it is quite challenging to obtain the labelled information for all the samples across all labels. Recently popular similarity-based classification approaches use label-label similarity and sample similarity to augment the incomplete labelled information. They employ the basic idea, i.e. similar instances should have similar labels and likewise, similar labels should have similar samples. Through this research, we establish a connection between the graph Laplacian-based approaches (used

commonly for incomplete supervision settings, e.g. refer [10]) and the similarity-based classification framework [8]. Further, we show that similarity-based learning frameworks are in fact an extension of the popular multi-label SVM [15]. The proposed model retains the best of the former two works (i.e. [8] and [15]). The implicit use of the data in the form of a similarity matrix grants robustness to the resulting models. The resulting proposed models are time efficient and showcase good performance. Further, the end product of the optimisation so obtained is often used as a regularisation term by researchers in [2, 12] to enhance the performance in limited supervision challenges in multi-label learning.

This work explores the application of similarity-based classification framework to the problem of Multilabel Learning with Missing Labels, wherein there may be 3 kinds of instances, i.e. instances with no missing labels (completely labelled samples), instances with incompletely labelled samples and instances with empty observed labels (unlabelled samples).

The present work is discussed in the following sections. Related work is reviewed in Sect. 2. Our proposed models are introduced in Sect. 3 and subsequently discusses the proposed model and its connection with the existing approaches. The experimental results are reported in Sect. 4, and finally, Sect. 5 concludes the paper.

2 Related Work

Consider a set of n instances $X = \{x_1, x_2, \dots, x_n\}$ each in a d -dimensional space, i.e., $x_i \in \mathbb{R}^d$, $i = 1, 2, \dots, n$ be associated with m labels. Thus, each sample x_i can take one or more labels from the m different labels c_1, \dots, c_m with its corresponding label vector $y_i \in \{1, 0, -1\}^m$, which determines its belonging to each of these labels. In general, the data set, $X \in \mathbb{R}^{n \times d}$, Y is the ground truth label matrix, $Y \in \{1, 0, -1\}^{n \times m}$. $Y_{i,j} = 1$ means that sample x_i is labelled as label c_j , similarly $Y_{i,j} = -1$ indicates that sample x_i is not labelled as c_j and $Y_{i,j} = 0$ denotes the missing information. Please note that in this paper the number of testing samples is denoted as n_t , and the number of training samples is denoted by n . $x_t \in \mathbb{R}^{n_t \times d}$, denotes the testing samples. A_s, A_c denote the sample and class similarity matrices, respectively. E is the matrix of all ones of the appropriate dimension, \circ denotes Hadamard product, \oslash denotes element-wise division and I is the identity matrix of the appropriate dimension. $\|\cdot\|_F$ indicates the frobenious norm, $\|\cdot\|$ denotes the absolute function

2.1 Multi-label Learning with Missing Labels (MLML) [10]

Authors in [10] aim to complete the missing information in the given ground truth label matrix Y by taking advantage of label consistency and label smoothness.

$$\min_Z \quad \|Z - Y\|_F^2 + \frac{c_1}{2} \text{Tr}(Z^T L_s Z) + \frac{c_2}{2} \text{Tr}(Z L_c Z^T) \quad (1)$$

Here, c_1, c_2 are the model parameters. Z is the prediction matrix having complete labels for all instances. L_s and L_c are the sample and label-based graph Laplacian matrix, respectively. The Laplacians are defined as:

$$\begin{aligned} L_s &= I - \alpha_s D_s^{-\frac{1}{2}} A_s D_s^{-\frac{1}{2}} \\ L_c &= I - \alpha_c D_c^{-\frac{1}{2}} A_c D_c^{-\frac{1}{2}} \end{aligned} \quad (2)$$

$$\begin{aligned} D_s &= I \circ (A_s E) \\ D_c &= I \circ (A_c E) \end{aligned} \quad (3)$$

The first term in the Eq. 1 enforces the label consistency, while the second and third terms implement label smoothness, i.e. sample-level smoothness and class-level smoothness, respectively. For constructing similarity matrix (A_s), arbitrary similarity functions like RBF kernel, linear kernel, etc. can be used. Since the label Y is discrete, diffusion kernels [5] are more suited for calculating label similarity.

The authors provide two solutions to Eq. 1

An exact solution using the Sylvester equation:

$$(I + c_1 L_s)Z + Z(c_2 L_c + I) = 2Y \quad (4)$$

An approximated solution:

$$Z = (1 - \alpha_s)(L_s)^{-1}Y(L_c)^{-1}(1 - \alpha_c) \quad (5)$$

Here, for Eq. 5 $\alpha_s = \frac{c_1}{c_1+1}$ and $\alpha_c = \frac{c_2}{c_2+1}$. Both solutions have cubic computational complexity. The aforementioned model uses the data information(i.e. X) implicitly in the form of sample graph Laplacian (L_s); hence it is less impacted by the outliers present in the data. Their model functions as a transductive model making it unsuitable for online learning extension. Moreover, their model's transductive nature and the cubic time complexity requirement make it unfavourable for large-scale applications. However, from the empirical observation of the results reported in their paper, their model can effectively handle the missing label problem in multi-label learning alongside exploiting the available label correlation.

2.2 Similarity-Based Multi-label Learning (SML) [8]

Authors in [8] introduced a novel Similarity-based approach for Multi-label Learning termed SML. The functioning of their model can be explained in terms of the following equation:

$$Z = A_s Y^+ \quad (6)$$

Here, $A_s \in \mathcal{R}^{n_t \times n}$ denotes the sample similarity matrix (i.e. similarity of n_t testing samples with the n training samples), $Y^+ = (|Y| + Y)/2$, $Z \in \mathcal{R}^{n_t \times m}$ contains the similarity score of testing samples for all labels. For a new test sample, the similarity score for a particular label is calculated by the cumulative similarity with the samples belonging to that label. Further, the test sample is assigned to a label if the similarity score exceeds a certain threshold (which is inferred from training data). Alternatively, authors have proposed a similarity-based label set prediction approach to predict the label set size.

SML falls under the category of lazy learners, with little or almost no training phase. Moreover, their model can be easily parallelised. This makes it suitable for large-scale applications. However, their model's performance directly depends on the thresholding function or label set size prediction. From the empirical observation of the results reported in their paper, it is evident that it outputs a sub-optimal prediction. Further, this model does not exploit the label-label correlation, which can enhance the prediction of a multi-label classifier.

2.3 Similarity-Based Multi-label Learning by Logistic Regression (SBLR) [7]

Authors in [7] employ Eq. 6 as a preprocessing technique. Further, they use logistic regression atop the output from the SML algorithm. In addition to exploiting the label dependencies, their model also avoids the impairment of the performance from noisy information by imposing L_1 -norm penalty into their model. The optimisation of their model can be written as:

$$\min_{\beta} \quad tr((Y^+)^T(\bar{Z}\beta)) + tr(E^T(\ln(E + \exp(\bar{Z}\beta)))) + \lambda \|\beta\|_1$$

Here, $E \in \{1\}^{n \times m}$, $\bar{Z} = [Z \quad 1]$, $Z = A_s Y^+$, $A_s \in \mathcal{R}^{n \times n}$, Y^+ have same meaning as described previously. λ is the model parameter used to control sparseness on β . Posterior probability p given Z_0 (as defined in Eq. 6) can be calculated as:

$$p = (E) \odot (E + \exp([Z_0 \quad 1]\beta))$$

Further, a testing sample is annotated 1 for a label if it has a posterior probability greater than 0.5 otherwise, 0. As mentioned in their paper, their approach has a complexity of $\mathcal{O}(nm^2 + dn^2)$.

3 Proposed Model

3.1 Formulation

Using $Tr(A^T A) = \|A\|_F^2$ (for some matrix A), the Eq. 1 can be rewritten as:

$$\min_Z \|Z - Y\|_F^2 + \frac{c_1}{2} \|L_s^{1/2} Z\|_F^2 + \frac{c_2}{2} \|Z L_c^{1/2}\|_F^2 \quad (7)$$

Z is minimised separately under the pre-multiplication and the post-multiplication of respective Laplacians in the second and third term, can be jointly minimised.

$$\left\| (L_s^{1/2})Z(L_c^{1/2}) \right\|_F^2 \simeq \left\| L_s^{1/2}Z \right\|_F^2 + \left\| ZL_c^{1/2} \right\|_F^2 \quad (8)$$

The resultant completed matrix $(L_s^{1/2}ZL_c^{1/2})$ should be consistent with the label matrix Y .

$$J = \min_Z \left\| (L_s^{1/2})Z(L_c^{1/2}) - Y \right\|_F^2 \quad (9)$$

Notice that Eq. 9 carries the essence of the Eq. 7, where the three terms in Eq. 7 are coheased into a single expression in Eq. 9 resulting in a parameter-free optimisation. Differentiating Eq. 9 w.r.t. Z we get

$$Z = (L_s^{-1/2})Y(L_c^{-1/2}) \quad (10)$$

The Eq. 10 is convex, but it has some limitations. Firstly it needs to invert two matrices of size $(n \times n)$ and $(m \times m)$, which has a cost of $\mathcal{O}(n^3 + m^3)$. Secondly, the Laplacian matrices and their matrix square root are positive semi-definite; hence their inverse does not exist. The Laplacian matrices can be modified to calculate the inverse as in Eq. 2. However, the modified Laplacian matrices lead to an approximate solution which can lead to sub-optimal prediction in certain cases.

The aforementioned reasons provide a need for alternatives to the Laplacian matrix. Laplacian is kind of a dissimilarity matrix, and its inverse is equivalent to similarity matrix. From the work in [9] (refer Theorem 1), it can be inferred that $L \simeq A^{-1}$ (or $L^{-1} \simeq A$). Taking this into account, Eq. 9 can be rewritten as:

$$J = \min_Z \left\| (A_s^{-1/2})Z(A_c^{-1/2}) - Y \right\|_F^2 \quad (11)$$

OR

$$J = \min_Z \left\| Z - (A_s^{1/2})Y(A_c^{1/2}) \right\|_F^2 \quad (12)$$

Differentiating Eq. 12 w.r.t. Z and rearranging we get

$$Z = (A_s^{1/2})Y(A_c^{1/2}) \quad (13)$$

The equation labeled as Eq. 12 represents the conclusive form of our proposed model. This model stands out due to its simplicity and robustness. The simplicity arises from its parameter-free nature, requiring only the input of two similarity matrices. Its robustness is evident as it utilizes the data X , explicitly in the form of the sample similarity matrix, thereby mitigating the impact of outliers. Consequently, we have named this model Simple Robust Multi-label Learning (SR-ML). Moreover, our model can handle the problem of Multi-label Learning

with Missing Labels effectively. Further, our model can be easily modified to suit different scenarios:

$$\begin{aligned}
Z_1 &= (A_s^{1/2})Y(A_c^{1/2}) & v_1 \\
Z_2 &= (A_s)Y(A_c) & v_2 \\
Z_3 &= (K(x_t, X)^{1/2})Y(A_c^{1/2}) & v_3 \\
Z_4 &= (K(x_t, X))Y(A_c) & v_4
\end{aligned} \tag{14}$$

The Eq. 14, provide different versions of our proposed model namely v_1 , v_2 , v_3 and v_4 . The v_1 and v_2 are transductive models which can be used to complete the incomplete label matrix Y . The v_3 and v_4 are lazy learners which can effectively leverage the missing or complete information available in Y and make a direct prediction on the testing data. The v_1 and v_3 have some extra time complexity to calculate matrix square roots via Eigenvalue decomposition and SVD, respectively. In fact, the complexity of v_1 is $\mathcal{O}(n^3 + m^3)$ and v_3 is $\mathcal{O}(m^2n + m^3)$. The version v_2 and v_4 only involve the matrix multiplication as a major step. Hence they have a complexity of $\mathcal{O}(n^2m + m^2n)$ and $\mathcal{O}(n_tmn + m^2n_t)$ respectively. The A_s and $A_s^{1/2}$ are both similarity matrices, hence in v_2 , v_4 , the square root of similarity matrices are omitted. The v_1 , v_3 have better results compared to v_2 , v_4 . However, the latter have lesser computational time costs; hence are better suitable for large-scale applications. The matrix square root step in a way shrinks and sphericalizes the data, which outputs better results.

3.2 Comparison with Existing Approaches

MLML [10] vs SR-ML: MLML is a transductive model that uses class and sample Laplacians to enforce label smoothness. In our case, we use sample and label similarity matrices to enforce label smoothness. Further, we combine all three terms in a single expression as in Eq. 11. The resulting optimisation solution Eq. 13 shares correspondence with the approximate solution to Eq. 1, wherein this solution (Eq. 5) requires calculating the inverse of two matrices. The other solution of MLML can be obtained via solving the Sylvester equation. It can be observed that both solutions of the MLML approach have a cubic time complexity. On the other hand, the solution to our model is simple and just requires the pre and post-multiplication of the sample and label similarity matrices with Y in the least case (i.e. v_2 and v_4). Further, our model is flexible and can be used as a transductive model for matrix completion or as a lazy learner for predicting the labels of test samples.

SML [8] vs SR-ML: SML takes into account only the relevant samples belonging to a label. It requires either a thresholding function or a label set size prediction to output the labels of the data. Meanwhile, SR-ML exploits the similarities with relevant and irrelevant samples, eliminating the need for thresholding. Further, our model makes use of label similarity matrix to incorporate label dependency into the model. Nevertheless, a thresholding function can be used as the last step to refine the prediction further.

Support Vector Machines for Multilabel/Multiclass Learning vs SR-ML. Authors in [4] provide a matrix formulation for SVM which optimises all classes/labels simultaneously via a single optimisation problem. The solution (Weight matrix W) to it can be written as:

$$W = X^T(\alpha \circ Y)A_c \quad (15)$$

$$XW = XX^T(\alpha \circ Y)A_c \quad (16)$$

$$Z = K(X, X) * (\alpha \circ Y)A_c \quad (17)$$

If $\alpha = E$, the above expression simplifies as:

$$Z = (K(X, X))Y(A_c) \quad (18)$$

which is same as v_2 of our proposed model. Hence it can be said that the similarity-based learning frameworks are in fact an extension of SVM.

3.3 Algorithm

The pseudo-code for the proposed algorithm SR-ML v_1, v_3 are presented in Algorithms 1 and 2. The pseudo-code for the remaining versions can be written on similar lines; we avoid writing it here for brevity.

Algorithm 1. SR-ML v1

Input:

- * Label similarity matrix: A_c , Dimension: $(n + n_t) \times (n + n_t)$
- * Sample similarity matrix: A_s , Dimension: $(m) \times (m)$

Compute Matrix square root:

▷ Using Eigenvalue Decomposition

$$A_s^{1/2} = P_s \Sigma_s^{1/2} P_s^T$$

$$A_c^{1/2} = P_c \Sigma_c^{1/2} P_c^T$$

Compute Complete Prediction Matrix

$$Z = A_s^{1/2} Y A_c^{1/2}$$

Output: Complete-Label-Matrix = $\text{sign}(Z)$

4 Experiments

The experiments are performed on well-known diverse datasets using ten-fold cross-validation, in MATLAB version 9.4 under Microsoft Windows environment on a machine with 16 GB RAM and 3.40 GHz i7 CPU. For generating missing labels, some (10%, 30%, 50% in Tables 2 and 3) percent of random elements in the target matrix Y are set to zero in each fold. The optimal value of user-defined parameters for different models are obtained by fine-tuning a validation

Algorithm 2. SR-ML v3**Input:*** Label similarity matrix: (A_c) , Dimension: $(m) \times (m)$ * Sample similarity matrix: $K(x_t, X)$, Dimension: $(n_t) \times (n)$ **Compute Matrix square root:**

▷ Using Eigenvalue Decomposition

$$A_c^{1/2} = P_c \Sigma_c^{1/2} P_c^T$$

Compute Matrix square root:

▷ Using SVD

$$K(x_t, X)^{1/2} = U_s \Sigma_s^{1/2} V_s^T$$

Compute Prediction For testing samples

$$Z = K(x_t, X)^{1/2} Y A_c^{1/2}$$

Output: Predicted-Labels = sign(Z)

set generated using ten percent of training data. After the model parameters are investigated, the validation set is sent back to training data for retraining. The range $([-1, 1])$ normalisation is performed on all datasets. We use Gaussian kernel $e^{-\rho \|x_i - x_j\|_2^2}$ for all to provide uniformity for comparison. ρ is the kernel parameter, for experiments we set $\rho = 0.01$. The time of the algorithms is measured in seconds using tic-toc (Matlab). The results are reported up to the third decimal; very small decimal values (e.g. 0.0001) are explicitly reported as zero. The best results are highlighted in bold. For LSML (Label-Specific features for multilabel learning with Missing Labels) [3], MLKNN (Multilabel learning lazy learning with K Nearest Neighbors) [14] and LMKNN (Large Margin metric learning with K Nearest Neighbors) [6] we use the parameter setting as described in their paper. The characteristics of the datasets (Instances \times features \times labels (domain)) used are described alongside the results.

Calculation of Similarity Matrices

For computing the sample similarity matrix A_s , we use the kernel function K . For computing the label similarity matrix A_c , we use diffusion kernels [13], which are better suited for discrete spaces [5]. The similarity between the two labels (c_i, c_k) is calculated using the Jaccard index.

4.1 Supervised Results on Datasets

The Table 1 shows the results for our proposed models (SR-ML) against the comparing models under the supervised setting. It can be seen that the SR-ML-v4 has the least time requirement, whereas the SR-ML-v3 outputs the best results in most cases. SBLR significantly improves upon the results as compared to SML with some increase in training time. The SR-ML models show better performance as they need not require a thresholding function as in SML. Further, it can be observed that SR-ML-v3 requires less time compared to SR-ML-v1, as the former has to deal with kernel matrix decomposition of a smaller size compared to the latter, though both have cubic training time complexity. It

Table 1. Supervised Results on Multi-label datasets

Emotions (593 × 72 × 6 (Music))											
	SR-ML-v1	SR-ML-v2	SR-ML-v3	SR-ML-v4	MLML-sylvester	MLML-approx	SML	SBLR	LSML	MLKNN	LMKNN
Time	0.028±0.001	0±0	0.009±0	0±0	0.159±0.006	0.006±0		0.026±0.001	0.315±0.012	0.006±0	0.069±0.001
Exact match	0.309±0.015	0.307±0.02	0.309±0.016	0.307±0.02	0.302±0.019	0.29±0.018		0.248±0.02	0.292±0.02	0.263±0.021	0.28±0.019
Hamming Loss	0.182±0.008	0.186±0.009	0.187±0.008	0.186±0.009	0.19±0.008	0.192±0.008		0.205±0.011	0.191±0.011	0.203±0.008	0.199±0.009
Macro F1	0.643±0.015	0.627±0.016	0.648±0.014	0.627±0.016	0.62±0.016	0.606±0.017		0.603±0.021	0.628±0.022	0.618±0.014	0.614±0.018
Micro F1	0.674±0.015	0.661±0.017	0.676±0.016	0.661±0.017	0.651±0.016	0.641±0.016		0.644±0.019	0.656±0.021	0.642±0.012	0.653±0.017
Avg Precision	0.764±0.015	0.759±0.017	0.763±0.015	0.759±0.017	0.752±0.018	0.751±0.017		0.739±0.016	0.741±0.018	0.739±0.012	0.742±0.013
Image (2000 × 294 × 5 (Image))											
Time	0.721±0.011	0.002±0	0.049±0.001	0±0	2.762±0.02	0.136±0.001		0.069±0.001	0.594±0.015	0.136±0.001	1.416±0.004
Exact match	0.393±0.013	0.347±0.01	0.451±0.013	0.347±0.01	0.327±0.01	0.449±0.014		0.31±0.011	0.34±0.01	0.355±0.005	0.415±0.012
Hamming Loss	0.167±0.005	0.176±0.004	0.161±0.005	0.176±0.004	0.179±0.004	0.169±0.005		0.194±0.005	0.182±0.004	0.186±0.004	0.17±0.004
Macro F1	0.558±0.011	0.504±0.01	0.616±0.009	0.504±0.01	0.488±0.01	0.603±0.011		0.516±0.01	0.536±0.009	0.547±0.008	0.583±0.008
Micro F1	0.562±0.012	0.513±0.011	0.613±0.01	0.513±0.011	0.498±0.011	0.602±0.011		0.517±0.011	0.536±0.01	0.549±0.008	0.587±0.01
Avg Precision	0.707±0.006	0.676±0.007	0.744±0.007	0.676±0.007	0.665±0.006	0.742±0.008		0.672±0.008	0.695±0.008	0.7±0.007	0.724±0.007
Scene (2407 × 294 × 6 (Image))											
Time	1.337±0.019	0.002±0	0.083±0.002	0±0	3±0.023	0.225±0.003		0.087±0.002	0.91±0.012	0.225±0.003	2.232±0.024
Exact match	0.59±0.007	0.576±0.008	0.651±0.009	0.576±0.008	0.577±0.006	0.639±0.009		0.402±0.009	0.449±0.008	0.451±0.011	0.629±0.008
Hamming Loss	0.092±0.002	0.093±0.002	0.093±0.003	0.093±0.002	0.092±0.002	0.103±0.003		0.12±0.002	0.111±0.002	0.117±0.002	0.087±0.002
Macro F1	0.7±0.008	0.691±0.008	0.732±0.008	0.691±0.008	0.696±0.007	0.712±0.009		0.57±0.007	0.606±0.008	0.61±0.009	0.736±0.006
Micro F1	0.701±0.007	0.693±0.007	0.725±0.008	0.693±0.007	0.696±0.006	0.704±0.008		0.583±0.007	0.611±0.007	0.611±0.008	0.732±0.005
Avg Precision	0.773±0.006	0.767±0.006	0.812±0.007	0.767±0.006	0.773±0.006	0.793±0.007		0.665±0.006	0.692±0.006	0.719±0.006	0.818±0.005
Yeast (2417 × 103 × 14 (Biology))											
Time	1.367±0.022	0.003±0	0.081±0.001	0±0	3.787±0.057	0.226±0.003		0.132±0.002	2.702±0.052	0.226±0.003	1.181±0.017
Exact match	0.193±0.007	0.188±0.007	0.206±0.006	0.188±0.007	0.189±0.007	0.184±0.007		0.135±0.006	0.177±0.009	0.147±0.008	0.184±0.007
Hamming Loss	0.187±0.002	0.188±0.002	0.186±0.002	0.188±0.002	0.187±0.002	0.188±0.002		0.203±0.002	0.196±0.002	0.199±0.002	0.192±0.002
Macro F1	0.372±0.007	0.368±0.008	0.4±0.008	0.368±0.008	0.37±0.007	0.367±0.007		0.311±0.005	0.405±0.009	0.349±0.004	0.388±0.007
Micro F1	0.644±0.004	0.641±0.005	0.654±0.004	0.641±0.005	0.645±0.005	0.642±0.005		0.607±0.005	0.647±0.005	0.636±0.004	0.649±0.004
Avg Precision	0.671±0.005	0.669±0.005	0.684±0.004	0.669±0.005	0.671±0.005	0.668±0.005		0.641±0.005	0.677±0.005	0.675±0.004	0.678±0.006
TMC-2007 (4000 × 500 × 22 (Text))											
Time	6.534±0.056	0.009±0	0.458±0.006	0.001±0	14.07±0.135	0.948±0.014		0.176±0.002	7.202±0.039	0.948±0.014	10.918±0.104
Exact match	0.202±0.006	0.2±0.005	0.231±0.007	0.2±0.005	0.213±0.005	0.218±0.006		0.075±0.004	0.133±0.005	0.245±0.008	0.209±0.007
Hamming Loss	0.077±0.001	0.074±0.001	0.094±0.003	0.074±0.001	0.075±0.001	0.083±0.001		0.093±0.001	0.086±0.001	0.066±0.001	0.073±0.001
Macro F1	0.273±0.007	0.272±0.009	0.334±0.01	0.272±0.009	0.298±0.006	0.369±0.008		0.129±0.009	0.159±0.009	0.464±0.01	0.299±0.009
Micro F1	0.535±0.005	0.54±0.005	0.53±0.009	0.54±0.005	0.566±0.004	0.553±0.004		0.235±0.007	0.445±0.007	0.633±0.005	0.594±0.006
Avg Precision	0.546±0.003	0.545±0.003	0.579±0.005	0.545±0.003	0.583±0.004	0.562±0.003		0.287±0.005	0.479±0.007	0.64±0.007	0.606±0.006

Table 2. Missing Label Results on TMC-2007 dataset

	SR-ML-v1	SR-ML-v2	SR-ML-v3	SR-ML-v4	MLML-sylvester	MLML-approx	SBLR	LSML
TMC-2007 (percentage of missing labels is 10)								
Exact match	0.192±0.005	0.189±0.005	0.205±0.005	0.189±0.005	0.203±0.004	0.197±0.006	0.121±0.003	0.232±0.008
Hamming Loss	0.077±0.001	0.074±0.001	0.1±0.003	0.074±0.001	0.076±0.001	0.084±0.001	0.087±0.001	0.068±0.001
Macro F1	0.276±0.007	0.272±0.008	0.32±0.008	0.272±0.008	0.288±0.007	0.349±0.006	0.141±0.009	0.454±0.008
Micro F1	0.536±0.005	0.539±0.006	0.513±0.008	0.539±0.006	0.559±0.003	0.538±0.004	0.422±0.006	0.622±0.005
Avg Precision	0.547±0.004	0.544±0.005	0.564±0.004	0.544±0.005	0.577±0.004	0.547±0.003	0.46±0.006	0.628±0.007
TMC-2007 (percentage of missing labels is 30)								
Exact match	0.173±0.003	0.171±0.004	0.163±0.006	0.171±0.004	0.183±0.003	0.152±0.005	0.041±0.002	0.203±0.006
Hamming Loss	0.079±0.001	0.076±0.001	0.115±0.003	0.076±0.001	0.078±0.001	0.088±0.001	0.096±0.001	0.074±0.001
Macro F1	0.265±0.006	0.258±0.007	0.296±0.007	0.258±0.007	0.281±0.005	0.316±0.009	0.087±0.006	0.436±0.007
Micro F1	0.529±0.004	0.532±0.003	0.475±0.007	0.532±0.003	0.549±0.002	0.504±0.005	0.128±0.004	0.598±0.004
Avg Precision	0.542±0.003	0.539±0.003	0.529±0.004	0.539±0.003	0.566±0.002	0.511±0.004	0.23±0.003	0.61±0.005
TMC-2007 (percentage of missing labels is 50)								
Exact match	0.155±0.006	0.153±0.006	0.115±0.007	0.153±0.006	0.165±0.006	0.126±0.004	0.02±0.003	0.168±0.005
Hamming Loss	0.081±0.001	0.078±0.001	0.141±0.003	0.078±0.001	0.08±0.001	0.091±0.001	0.099±0.001	0.085±0.001
Macro F1	0.261±0.006	0.254±0.007	0.261±0.003	0.254±0.007	0.27±0.007	0.295±0.007	0.062±0.004	0.389±0.008
Micro F1	0.521±0.006	0.525±0.006	0.425±0.005	0.525±0.006	0.537±0.003	0.479±0.004	0.073±0.004	0.558±0.004
Avg Precision	0.537±0.006	0.535±0.006	0.488±0.005	0.535±0.006	0.556±0.004	0.484±0.004	0.198±0.003	0.577±0.006

can be easily observed that our proposed models (all versions) show closely similar results to MLML models, strengthening our claim that our model forms a connecting link between MLML [10] and SML [8].

Table 3. Missing Label Results on Emotions, Image, Scene, and Yeast datasets

	SR-ML-v1	SR-ML-v2	SR-ML-v3	SR-ML-v4	MLML-sylvester	MLML-approx	SBLR	LSML
Emotions (percentage of missing labels is 10)								
Exact match	0.298±0.019	0.298±0.022	0.288±0.017	0.298±0.022	0.295±0.02		0.288±0.017	0.248±0.019
Hamming Loss	0.187±0.008	0.189±0.008	0.19±0.008	0.189±0.008	0.19±0.009		0.191±0.008	0.203±0.01
Macro F1	0.633±0.015	0.622±0.015	0.644±0.014	0.622±0.015	0.615±0.018		0.605±0.018	0.578±0.02
Micro F1	0.663±0.016	0.653±0.016	0.67±0.015	0.653±0.016	0.648±0.018		0.639±0.017	0.602±0.02
Avg Precision	0.76±0.016	0.759±0.017	0.762±0.015	0.759±0.017	0.756±0.018		0.755±0.016	0.711±0.017
Emotions (percentage of missing labels is 30)								
Exact match	0.297±0.018	0.295±0.021	0.285±0.019	0.295±0.021	0.294±0.024		0.282±0.021	0.128±0.018
Hamming Loss	0.191±0.007	0.19±0.008	0.199±0.008	0.19±0.008	0.19±0.009		0.192±0.008	0.245±0.01
Macro F1	0.632±0.012	0.629±0.015	0.635±0.013	0.629±0.015	0.623±0.017		0.606±0.016	0.388±0.028
Micro F1	0.66±0.014	0.656±0.016	0.66±0.016	0.656±0.016	0.653±0.018		0.643±0.017	0.41±0.028
Avg Precision	0.755±0.017	0.758±0.019	0.751±0.014	0.758±0.019	0.754±0.019		0.752±0.018	0.643±0.02
Emotions (percentage of missing labels is 50)								
Exact match	0.275±0.015	0.272±0.012	0.248±0.014	0.272±0.012	0.27±0.016		0.261±0.017	0.027±0.008
Hamming Loss	0.197±0.008	0.197±0.007	0.206±0.007	0.197±0.007	0.194±0.008		0.191±0.008	0.291±0.005
Macro F1	0.618±0.015	0.61±0.016	0.616±0.013	0.61±0.016	0.607±0.018		0.598±0.018	0.133±0.018
Micro F1	0.65±0.015	0.643±0.016	0.645±0.013	0.643±0.016	0.643±0.017		0.633±0.018	0.135±0.019
Avg Precision	0.741±0.017	0.736±0.02	0.731±0.016	0.736±0.02	0.743±0.02		0.739±0.02	0.564±0.016
Image (percentage of missing labels is 10)								
Exact match	0.387±0.013	0.343±0.011	0.442±0.014	0.343±0.011	0.322±0.012		0.438±0.016	0.306±0.008
Hamming Loss	0.169±0.005	0.178±0.004	0.164±0.005	0.178±0.004	0.18±0.004		0.174±0.006	0.187±0.004
Macro F1	0.555±0.011	0.504±0.01	0.612±0.01	0.504±0.01	0.486±0.01		0.592±0.012	0.489±0.011
Micro F1	0.559±0.011	0.512±0.012	0.609±0.011	0.512±0.012	0.495±0.011		0.591±0.013	0.485±0.011
Avg Precision	0.706±0.006	0.674±0.008	0.737±0.008	0.674±0.008	0.665±0.007		0.738±0.008	0.672±0.008
Image (percentage of missing labels is 30)								
Exact match	0.369±0.013	0.328±0.012	0.413±0.014	0.328±0.012	0.316±0.012		0.412±0.017	0.234±0.008
Hamming Loss	0.176±0.005	0.182±0.005	0.176±0.005	0.182±0.005	0.183±0.005		0.18±0.006	0.198±0.004
Macro F1	0.536±0.012	0.49±0.012	0.589±0.01	0.49±0.012	0.48±0.012		0.577±0.014	0.383±0.01
Micro F1	0.539±0.013	0.498±0.013	0.587±0.011	0.498±0.013	0.489±0.013		0.577±0.015	0.385±0.01
Avg Precision	0.693±0.007	0.664±0.008	0.73±0.008	0.664±0.008	0.657±0.007		0.72±0.01	0.629±0.009
Image (percentage of missing labels is 50)								
Exact match	0.384±0.01	0.336±0.007	0.394±0.015	0.336±0.007	0.32±0.009		0.401±0.013	0.148±0.007
Hamming Loss	0.176±0.004	0.185±0.004	0.186±0.006	0.185±0.004	0.186±0.004		0.184±0.005	0.213±0.003
Macro F1	0.544±0.007	0.494±0.007	0.575±0.01	0.494±0.007	0.481±0.008		0.571±0.009	0.27±0.007
Micro F1	0.549±0.007	0.502±0.008	0.575±0.01	0.502±0.008	0.491±0.008		0.574±0.01	0.271±0.007
Avg Precision	0.696±0.005	0.667±0.007	0.717±0.007	0.667±0.007	0.66±0.007		0.714±0.007	0.59±0.009
Scene (percentage of missing labels is 10)								
Exact match	0.587±0.007	0.57±0.009	0.621±0.009	0.57±0.009	0.569±0.007		0.611±0.009	0.394±0.009
Hamming Loss	0.092±0.002	0.094±0.002	0.098±0.003	0.094±0.002	0.094±0.002		0.105±0.003	0.118±0.002
Macro F1	0.7±0.007	0.688±0.009	0.719±0.008	0.688±0.009	0.692±0.007		0.702±0.008	0.546±0.011
Micro F1	0.7±0.007	0.69±0.008	0.713±0.008	0.69±0.008	0.692±0.006		0.694±0.008	0.554±0.01
Avg Precision	0.775±0.007	0.766±0.007	0.799±0.008	0.766±0.007	0.771±0.006		0.784±0.007	0.654±0.007
Scene (percentage of missing labels is 30)								
Exact match	0.573±0.007	0.556±0.006	0.566±0.009	0.556±0.006	0.56±0.006		0.584±0.008	0.269±0.007
Hamming Loss	0.095±0.002	0.098±0.001	0.113±0.002	0.098±0.001	0.095±0.002		0.106±0.002	0.136±0.002
Macro F1	0.693±0.008	0.679±0.007	0.685±0.007	0.679±0.007	0.687±0.007		0.696±0.009	0.412±0.008
Micro F1	0.691±0.007	0.679±0.006	0.678±0.006	0.679±0.006	0.686±0.006		0.687±0.008	0.418±0.009
Avg Precision	0.769±0.006	0.76±0.005	0.779±0.007	0.76±0.005	0.769±0.006		0.773±0.007	0.582±0.007
Scene (percentage of missing labels is 50)								
Exact match	0.555±0.007	0.54±0.007	0.496±0.005	0.54±0.007	0.543±0.007		0.526±0.007	0.146±0.007
Hamming Loss	0.099±0.002	0.1±0.002	0.134±0.002	0.1±0.002	0.099±0.002		0.118±0.002	0.155±0.001
Macro F1	0.679±0.006	0.667±0.007	0.64±0.005	0.667±0.007	0.673±0.006		0.668±0.006	0.25±0.01
Micro F1	0.679±0.006	0.668±0.006	0.635±0.005	0.668±0.006	0.674±0.005		0.659±0.006	0.257±0.009
Avg Precision	0.757±0.005	0.747±0.006	0.743±0.005	0.747±0.006	0.754±0.005		0.746±0.005	0.516±0.006
Yeast (percentage of missing labels is 10)								
Exact match	0.185±0.008	0.185±0.008	0.196±0.005	0.185±0.008	0.185±0.007		0.184±0.007	0.157±0.008
Hamming Loss	0.188±0.002	0.189±0.002	0.187±0.001	0.189±0.002	0.188±0.002		0.189±0.002	0.198±0.002
Macro F1	0.371±0.008	0.369±0.008	0.4±0.007	0.369±0.008	0.369±0.007		0.366±0.007	0.378±0.008
Micro F1	0.642±0.005	0.641±0.005	0.653±0.003	0.641±0.005	0.644±0.004		0.642±0.004	0.627±0.005
Avg Precision	0.67±0.005	0.668±0.005	0.684±0.004	0.668±0.005	0.671±0.005		0.669±0.005	0.661±0.005
Yeast (percentage of missing labels is 30)								
Exact match	0.175±0.006	0.177±0.006	0.188±0.005	0.177±0.006	0.177±0.006		0.178±0.006	0.078±0.004
Hamming Loss	0.189±0.002	0.19±0.002	0.19±0.002	0.19±0.002	0.189±0.002		0.189±0.002	0.23±0.01
Macro F1	0.379±0.007	0.369±0.007	0.4±0.007	0.369±0.007	0.369±0.007		0.366±0.007	0.277±0.007
Micro F1	0.641±0.004	0.641±0.004	0.649±0.004	0.641±0.004	0.643±0.005		0.642±0.004	0.456±0.006
Avg Precision	0.67±0.005	0.669±0.005	0.682±0.004	0.669±0.005	0.67±0.005		0.669±0.005	0.582±0.006
Yeast (percentage of missing labels is 50)								
Exact match	0.16±0.007	0.163±0.007	0.165±0.006	0.163±0.007	0.163±0.009		0.163±0.009	0.016±0.002
Hamming Loss	0.193±0.002	0.194±0.002	0.197±0.002	0.194±0.002	0.192±0.002		0.192±0.002	0.275±0.001
Macro F1	0.371±0.007	0.37±0.007	0.402±0.006	0.37±0.007	0.368±0.006		0.364±0.006	0.143±0.004
Micro F1	0.636±0.005	0.635±0.005	0.641±0.003	0.635±0.005	0.637±0.005		0.637±0.005	0.198±0.004
Avg Precision	0.667±0.005	0.666±0.005	0.677±0.005	0.666±0.005	0.668±0.005		0.666±0.005	0.489±0.004

4.2 Missing Label Results on Datasets

The Tables 2 and 3 shows the results for our proposed models (SR-ML) against the comparing models under the missing label setting. It can be observed that when the missing label percentage is increased, the results for the proposed model and MLML models show a gradual decline, whereas the SBLR approaches show a sharp decline in their results. Overall, the SR-ML-v3 outputs the best results in most cases, otherwise close to the best results in some cases.

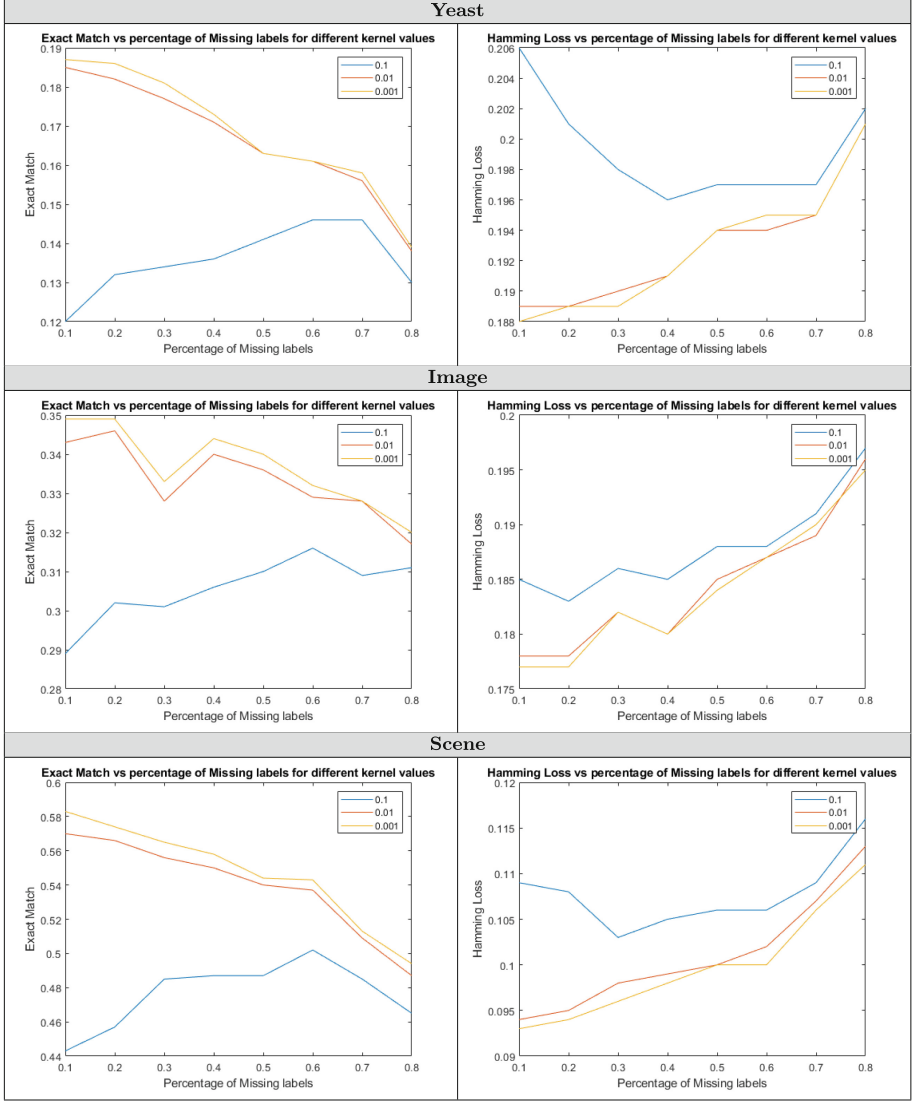
4.3 Ablation Study

In this section, we explore the effect of kernel parameter ρ across different percentages of missing labels for different datasets in Table 4 for SRML v₂. From the empirical observation of results reported in Table 4, it can be noticed that the SRML v₂ attains better results for lower kernel values (i.e. 0.01 and 0.001), however there seems not significant difference in results among smaller ρ values (0.01, 0.001). It can be noticed that similar trend follows for different datasets. Hence smaller ρ (0.01) value is suitable for SRML

5 Conclusions

In this paper, we have proposed SR-ML to address the problem of Multi-label Learning with Missing Labels effectively. From the theoretical and empirical observation of the results mentioned in this paper, we bridge the gap between the popularly used graph Laplacian [10] and the Similarity-Based Learning methods for Multi-label Learning [8]. To add more, we have shown that the proposed models share a connection with the SVM [15]. Through this research, we propose a simple, generic and efficient optimization to tackle the Multi-label learning problem with Missing Labels. However, the design of our model can be easily improved or extended to handle other limited supervision challenges in multi-label learning, Partial Multi-label Learning [11], Semi-Supervised Weak label Learning (SSWL) [2], for example. In this paper, we present four variants of SR-ML, two transductive and two lazy learning methods, which find applicability as per the convenience of the user/situation. In our opinion, lazy learners can be used when prediction is required only for testing samples, whereas transductive learners could be used in the case of matrix completion problems. We will look towards the latter in our future research. We highlight the promising potential of models such as SR-ML and the approach put forth in [1], which are characterized by their simplicity and reliance on SVM principles for making explicit predictions. Further research needs to be conducted to expand the understanding of such models.

Theorem 1. *The inverse of the Laplacian matrix (L), denoted as L^{-1} , is a kernel [9].*

Table 4. Effect of ρ for different percentages of missing labels for Multi-label datasets

Proof. Let the each element of L^{-1} , be defined as $l_{i,j}$, e_i be the standard basis column vector.

$$\begin{aligned}
 l_{i,j} &= e_i^T L^{-1} e_j \\
 &= e_i^T U \Sigma U^T e_j \text{ (By using Eigenvalue decomposition)} \\
 &= (e_i^T U \Sigma^{1/2}) (\Sigma^{1/2} U^T e_j) \\
 &= (\Sigma^{1/2} x_i)^T (\Sigma^{1/2} x_j) \\
 &= x_i' x_j^T
 \end{aligned}$$

Thus we can express $L^{-1} = X'(X')^T = \phi(X')\phi((X')^T) = K(X', (X')^T)$, where X' denotes the data matrix, each row representing a sample, K denotes kernel function.

For more details, please refer to [9]

References

1. Akram-Ali-Hammouri, Z., Fernández-Delgado, M., Cernadas, E., Barro, S.: Fast support vector classification for large-scale problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(10), 6184–6195 (2021)
2. Dong, H.C., Li, Y.F., Zhou, Z.H.: Learning from semi-supervised weak-label data. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
3. Huang, J., et al.: Improving multi-label classification with missing labels by learning label-specific features. *Inf. Sci.* **492**, 124–146 (2019)
4. Jain Reshma Rastogi, S.: Enhancing pattern classification in support vector machines through matrix formulation. arXiv e-prints pp. arXiv–2307 (2023)
5. Kondor, R., Lafferty, J.: Diffusion kernels on graphs and other discrete input spaces. In: *Proceedings of ICML 2002*, pp. 315–322 (2002)
6. Liu, W., Tsang, I.: Large margin metric learning for multi-label prediction. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29 (2015)
7. Ma, Z., Chen, S.: A similarity-based framework for classification task. *IEEE Trans. Knowl. Data Eng.* **35**, 5438–5443 (2022)
8. Rossi, R.A., Ahmed, N.K., Eldardiry, H., Zhou, R.: Similarity-based multi-label learning. In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE (2018)
9. Saerens, M., Fous, F., Yen, L., Dupont, P.: The principal components analysis of a graph, and its relationships to spectral clustering. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) *ECML 2004. LNCS (LNAI)*, vol. 3201, pp. 371–383. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30115-8_35
10. Wu, B., Liu, Z., Wang, S., Hu, B.G., Ji, Q.: Multi-label learning with missing labels. In: *2014 22nd International Conference on Pattern Recognition*, pp. 1964–1968. IEEE (2014)
11. Xie, M.K., Huang, S.J.: Partial multi-label learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
12. Xie, M.K., Huang, S.J.: Semi-supervised partial multi-label learning. In: *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 691–700. IEEE (2020)
13. Zaki, M.J., Meira, W., Jr.: *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge University Press, Cambridge (2020)
14. Zhang, M.L., Zhou, Z.H.: MI-knn: a lazy learning approach to multi-label learning. *Pattern Recogn.* **40**(7), 2038–2048 (2007)
15. Zhang, Y., Yeung, D.Y.: Multilabel relationship learning. *ACM Trans. Knowl. Disc. Data (TKDD)* **7**(2), 1–30 (2013)



GL-TSVM: A Robust and Smooth Twin Support Vector Machine with Guardian Loss Function

Mushir Akhtar, M. Tanveer^(✉), and Mohd. Arshad

Indian Institute of Technology Indore, Simrol, Indore, India
{phd2101241004,mtanveer,arshad}@iiti.ac.in

Abstract. Twin support vector machine (TSVM), a variant of support vector machine (SVM), has garnered significant attention due to its 3/4 times lower computational complexity compared to SVM. However, due to the utilization of the hinge loss function, TSVM is sensitive to outliers or noise. To remedy it, we introduce the guardian loss (G-loss), a novel loss function distinguished by its asymmetric, bounded, and smooth characteristics. We then fuse the proposed G-loss function into the TSVM and yield a robust and smooth classifier termed GL-TSVM. Further, to adhere to the structural risk minimization (SRM) principle and reduce overfitting, we incorporate a regularization term into the objective function of GL-TSVM. To address the optimization challenges of GL-TSVM, we devise an efficient iterative algorithm. The experimental analysis on UCI and KEEL datasets substantiates the effectiveness of the proposed GL-TSVM in comparison to the baseline models. Moreover, to showcase the efficacy of the proposed GL-TSVM in the biomedical domain, we evaluated it on the breast cancer (BreCaHis) and schizophrenia datasets. The outcomes strongly demonstrate the competitiveness of the proposed GL-TSVM against the baseline models. The supplementary file, along with the source code for the proposed GL-TSVM model, is publicly accessible at <https://github.com/mtanveer1/GL-TSVM>.

Keywords: Support vector machine · Twin support vector machine · Robust classification · Asymmetric loss function · Iterative algorithm

1 Introduction

Support vector machine (SVM) [5], a kernel-based method, has been extensively researched over the last two decades, particularly in the realm of pattern recognition. It is rooted in the concept of structural risk minimization (SRM) and is derived from statistical learning theory (SLT), consequently having a solid theoretical foundation and demonstrating better generalization capabilities. Its wide-ranging applications span across diverse domains, including cancer diagnosis [14], Alzheimer detection [16], and so forth.

The key idea of SVM revolves around identifying two parallel hyperplanes with the maximum possible margin between them. This primarily involves solving a quadratic programming problem (QPP) whose complexity is proportional

to the cube of the training dataset size. Twin SVM (TSVM) [11], a variant of SVM, tackles this problem by solving two smaller QPPs instead of one large QPP, thereby reducing computational costs by approximately 75% compared to traditional SVM methods. This significant computational efficiency has garnered considerable attention from the research community, leading to extensive studies aimed at enhancing the performance of TSVM. For instance, Kumar and Gopal [12] proposed the least squares TSVM, which stands out for its simplicity and efficiency in binary classification tasks. Additionally, Shao et al. [18] introduced another improved variant, the twin bounded SVM, which boosts the generalization capabilities of TSVM through the incorporation of structural risk minimization term. Besides, numerous researchers have developed several TSVM variants to further enhance its performance across different applications. As an example, to tackle the imbalance problem, Ganaie et al. [9] proposed the large-scale fuzzy least squares TSVM. To develop a robust and sparse variant of TSVM, Tanveer [21] reformulated the classical TSVM by incorporating a regularization technique and proposed an exact 1-norm linear programming formulation for TSVM. To address the noise sensitivity of TSVM, Tanveer et al. [26] proposed the large-scale pin-TSVM by utilizing the pinball loss function. Both of the aforementioned algorithms eliminate the requirement of matrix inversion, making them suitable for large-scale problems. To delve deeper into the development of TSVM models, readers can refer to [22, 25].

Despite several strengths of TSVM, it still has opportunities for improvement. One key area is its sensitivity to outliers or noise, which stems from the unbounded escalation of the hinge loss function, leading to excessively high losses for samples located far from the proximal hyperplane [27]. Also, the hinge loss function solely imposes penalties on misclassified samples and neglects the contribution of the correctly classified samples. However, the influence of samples from distinct classes, positioned on either side of the hyperplane, on the decision hyperplane differs based on their respective locations. Various researchers have conducted extensive investigations to reduce the susceptibility of TSVM to noise and outliers [8, 19]. Among these approaches, designing a robust loss function has emerged as a crucial focus. Recent advancements have introduced several robust loss functions for TSVM, such as the symmetric LINEX loss function [19], pinball loss function [26], correntropy-induced loss function [28], Huber loss function [4], and others. Despite their contributions to enhancing the robustness of TSVM, these methods still present notable limitations. The symmetric LINEX loss and Huber loss functions, while innovative, suffer from symmetry and unbounded growth. Their unbounded nature makes them vulnerable to extreme values, thereby increasing sensitivity to outliers or noise. Additionally, their symmetric design treats samples on either side of the proximal hyperplane equally, potentially ignoring the differing influences these samples exert on the decision boundary. The correntropy-induced loss, though bounded, maintains symmetry, similarly equalizing the importance of samples regardless of their positioning. Conversely, the pinball loss function introduces asymmetry but lacks boundedness, failing to cap the impact of extreme data points. In essence, although these

loss functions mark significant strides in the robustness of TSVM, their inherent limitations highlight the ongoing need for further research. Developing more effective and versatile loss functions is essential to advancing robust machine learning models, capable of better handling outliers and noise, thereby ensuring more reliable and accurate predictions.

Taking motivation from prior research, in this paper, we develop a robust loss function, named guardian loss (G-loss). It is meticulously designed to possess asymmetric, bounded, and smooth characteristics. Then, we amalgamate the proposed G-loss function into TSVM and introduce a robust and smooth classifier termed GL-TSVM. The main contributions of this paper can be outlined as follows:

1. To shield TSVM against outliers or noise, we introduce the guardian loss function (G-loss), a novel approach characterized by its asymmetry, boundedness, and smoothness. The asymmetric feature of G-loss enables the assignment of distinct penalties to distinct samples based on their location with respect to the proximal hyperplane. The bounded nature allows for a strict limit on the maximum loss for data points with significant deviations, thereby mitigating the influence of noise or outliers. Furthermore, the smoothness property empowers the utilization of gradient-based algorithms for model optimization.
2. We incorporate the G-loss function into TSVM and propose a novel robust and smooth classifier named GL-TSVM. Additionally, we devised an iterative algorithm to address the optimization problems of GL-TSVM.
3. To employ the principle of structural risk minimization (SRM) and avoid the overfitting problem, we introduce a regularization term into the objective function of GL-TSVM.
4. We perform numerical evaluation on benchmark UCI and KEEL datasets from various domains. The outcomes reveal the superior performance of the proposed GL-TSVM against the baseline models.
5. To evaluate the superiority of the proposed GL-TSVM in the biomedical realm, we conducted experiments using the breast cancer (BreCaKHis) and schizophrenia datasets. The results provide compelling evidence of the proposed GL-TSVM applicability in the biomedical domain.

The rest of this paper is structured as follows: The related works are discussed briefly in Sect. 2. Section 3 presents the proposed G-loss function and provides the formulation of GL-TSVM. Section 4 showcases the results of the experiments. Lastly, Sect. 5 concludes the paper with future directions.

2 Related Work

This section begins by defining the notations employed in this paper. Following this, we provide a brief overview of some relevant loss functions. The formulation of TSVM is briefly discussed in Section S.I of the supplement file.

2.1 Notations

Consider the training set denoted as $\{x_i, y_i\}_{i=1}^l$, where $x_i \in \mathbb{R}^n$ represents the sample vector and $y_i \in \{-1, 1\}$ signifies the corresponding class label. Let $X_+ = (x_1, \dots, x_{l_+})^\top \in \mathbb{R}^{l_+ \times n}$ and $X_- = (x_1, \dots, x_{l_-})^\top \in \mathbb{R}^{l_- \times n}$ represent matrices containing positive and negative instances, where l_+ and l_- denote the count of positive and negative instances, respectively, and $l = l_+ + l_-$. Further, e_1 and e_2 are identity vectors of appropriate size, and I is the identity matrix of appropriate size.

2.2 Loss Functions

In this subsection, we review some relevant loss functions, chosen to provide motivation for the work presented in this paper. Further, we provide the visual representation of baseline loss function in Fig. 1.

1. **Pinball loss function:** To improve the efficacy of TSVM against noise, Tanveer et al. [23] proposed TSVM with pinball loss function (Pin-GTSVM). The mathematical formulation of the pinball loss function is expressed as:

$$\mathcal{L}_{pin}(r) = \begin{cases} r, & r > 0, \\ -\tau r, & r \leq 0, \end{cases} \quad (1)$$

where $r = 1 - yf(x)$ and $\tau \in [0, 1]$. For $\tau = 0$, it reduces to the hinge loss function. It is asymmetric, unbounded, and non-smooth.

2. **Huber loss function:** To enhance the robustness of TSVM, Borah and Gupta [4] incorporated the Huber loss function in to TSVM. It is a combination of quadratic and linear loss. The mathematical representation of the Huber loss function is articulated as follows:

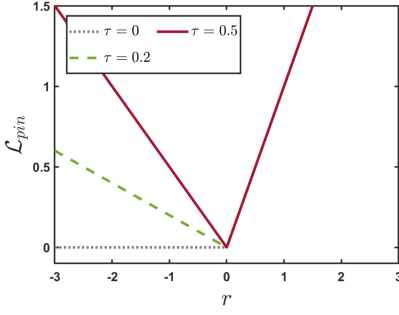
$$\mathcal{L}_{Huber}(r) = \begin{cases} \frac{1}{2}r^2, & |r| \leq \theta, \\ \theta|r| - \frac{1}{2}\theta^2, & \text{otherwise,} \end{cases} \quad (2)$$

where θ is a trade-off parameter between quadratic and linear loss. It is symmetric, unbounded, and smooth.

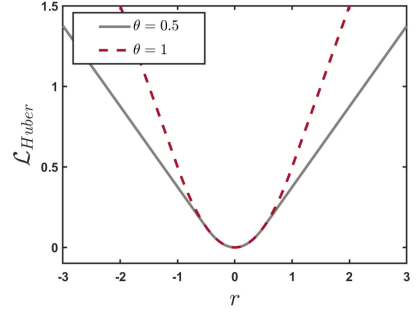
3. **Correntropy-induced loss function:** To enhance the robustness of TSVM against outliers, Zheng et al. [28] introduced the correntropy-induced loss into TSVM and proposed a robust TSVM model. The mathematical expression for the correntropy-induced loss is given as:

$$\mathcal{L}_{ce}(r) = \lambda \left[1 - \exp\left(\frac{-r^2}{\rho^2}\right) \right], \quad \forall r \in \mathbb{R}, \quad (3)$$

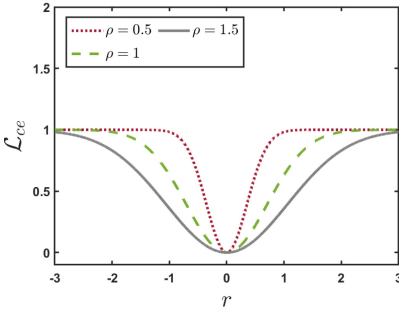
where ρ is the normalizing constant and $\lambda > 0$ is the loss parameter. It is symmetric, bounded, and smooth.



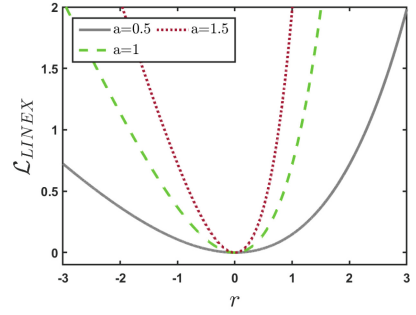
(a)



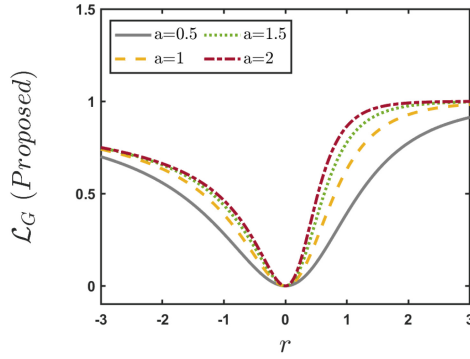
(b)



(c)



(d)



(e)

Fig. 1. Visual illustration of baseline and proposed G-loss function. (a) Pinball loss function with $\tau = 0$, $\tau = 0.2$, and $\tau = 0.5$. (b) Huber loss function with $\theta = 0.5$ and $\theta = 1$. (c) Correntropy-induced loss function with $\rho = 0.5$, $\rho = 1$, and $\rho = 1.5$. (d) LINEX loss function with $a = 0.5$, $a = 1$, and $a = 1.5$. (e) Proposed G-loss function with $a = 0.5$, $a = 1$, $a = 1.5$, and $a = 2$.

4. **LINEX loss function:** To advance TSVM against noise or outliers, Si et al. [19] incorporated the LINEX loss into TSVM. The mathematical representation of the LINEX loss is as follows:

$$\mathcal{L}_{LINEX}(r) = \exp(ar) - ar - 1, \quad \forall r \in \mathbb{R}, \quad (4)$$

where $a \neq 0$ is the loss parameter that controls the penalty for classified and misclassified samples. Si et al. [19] utilized the linear subpart of LINEX loss and introduced the symmetric LINEX loss into TSVM. It is unbounded and smooth.

In addition to these, recent advancements in designing robust and smooth functions include RoBoSS loss [1], Wave loss [3, 17], HawkEye loss [2], and so forth.

3 Proposed Work

In this section, we present a novel advancement, the guardian loss function (G-loss), designed to shield the supervised algorithm against outliers or noise. Then, we amalgamate the proposed guardian loss function into TSVM and propose a novel robust and smooth classifier coined GL-TSVM.

3.1 Guardian Loss Function

In this subsection, we introduce a novel loss function, the guardian loss (G-loss), designed to guide and fortify traditional algorithms against outliers and noise. It is meticulously designed to manifest asymmetry, boundedness, and smooth characteristics. The mathematical formulation of the G-loss function is as follows:

$$\begin{aligned} \mathcal{L}_G(r) &= \frac{r\{\exp(ar) - 1\}}{1 + r\{\exp(ar) - 1\}}, \\ &= 1 - \frac{1}{1 + r\{\exp(ar) - 1\}}, \quad \forall r \in \mathbb{R}, \end{aligned} \quad (5)$$

where $r = 1 - yf(x)$ and $a > 0$ is the parameter that governs the asymmetry of the G-loss function. Figure 1e showcases the visual representation of the G-loss function for varying values of a . Its asymmetric, bounded, and smooth nature serves to guide the algorithm in the right direction during the training process; hence, it is named the guardian loss. The asymmetric feature of it allows for distinct penalties for distinct samples based on their positioning relative to the proximal hyperplane. Its bounded nature allows it to impose a strict limit on the maximum loss for data points with significant deviations, thereby mitigating the influence of noise or outliers. Furthermore, the smoothness property empowers the utilization of gradient-based iterative algorithms for model optimization.

3.2 Linear GL-TSVM

Given a binary training dataset, the objective of linear GL-TSVM is to seek positive and negative hyperplanes as follows:

$$f_+ = u_+^\top x + b_+ = 0 \quad \text{and} \quad f_- = u_-^\top x + b_- = 0, \quad (6)$$

where $u_+, u_- \in \mathbb{R}^n$ and $b_+, b_- \in \mathbb{R}$ are the model parameters. To obtain the hyperplanes (6), we formed the primal problem of linear GL-TSVM as follows: (Linear GL-TSVM-1)

$$\begin{aligned} \min_{u_+, b_+, \zeta_-} \quad & \frac{1}{2} \sum_{i=1}^{l_+} (u_+^\top x_i + b_+)^2 + \frac{1}{2} C_1 (\|u_+\|_2^2 + b_+^2) + C_2 \sum_{j=1}^{l_-} \zeta_j^-, \\ \text{s.t.} \quad & \zeta_j^- = 1 - \frac{1}{1 + (1 + u_+^\top x_j + b_+) [\exp\{a(1 + u_+^\top x_j + b_+)\} - 1]}, \quad j = 1, \dots, l_-, \end{aligned} \quad (7)$$

(Linear GL-TSVM-2)

$$\begin{aligned} \min_{u_-, b_-, \zeta_+} \quad & \frac{1}{2} \sum_{j=1}^{l_-} (u_-^\top x_j + b_-)^2 + \frac{1}{2} C_3 (\|u_-\|_2^2 + b_-^2) + C_4 \sum_{i=1}^{l_+} \zeta_i^+, \\ \text{s.t.} \quad & \zeta_i^+ = 1 - \frac{1}{1 + (1 - u_-^\top x_i - b_-) [\exp\{a(1 - u_-^\top x_i - b_-)\} - 1]}, \quad i = 1, \dots, l_+, \end{aligned} \quad (8)$$

where $\zeta_+ = (\zeta_1^+, \dots, \zeta_{l_+}^+)^\top \in \mathbb{R}^{l_+}$, $\zeta_- = (\zeta_1^-, \dots, \zeta_{l_-}^-)^\top \in \mathbb{R}^{l_-}$. To be concise, we solely discuss the optimization problem (7), with the understanding that optimization problem (8) follows a similar structure. The objective function outlined in Eq. (7) comprises three distinct components. The first component aims to minimize the distance between the positive hyperplane and the positive instances. The second component, a regularization term, is included to adhere to the structural risk minimization principle. Lastly, the third component accounts for the cumulative penalty of all negative samples by leveraging the proposed G-loss function. Due to the non-convex nature of optimization problems (7) and (8), and utilizing their inherent smoothness property, we devised an iterative algorithm to solve them. Initially, we convert (7) and (8) into vector-matrix form as follows:

$$\min_{u_1} Q_1(u_1) = \frac{1}{2} \|M^\top u_1\|_2^2 + \frac{1}{2} C_1 \|u_1\|_2^2 + C_2 \mathcal{L}_1(u_1), \quad (9)$$

$$\min_{u_2} Q_2(u_2) = \frac{1}{2} \|N^\top u_2\|_2^2 + \frac{1}{2} C_3 \|u_2\|_2^2 + C_4 \mathcal{L}_2(u_2), \quad (10)$$

where $\mathcal{L}_1(u_1) = \sum_{j=1}^{l_-} 1 - \frac{1}{1 + (1 + N_j^\top u_1) [\exp\{a(1 + N_j^\top u_1)\} - 1]}$, $j = 1, \dots, l_-$; $\mathcal{L}_2(u_2) = \sum_{i=1}^{l_+} 1 - \frac{1}{1 + (1 - M_i^\top u_2) [\exp\{a(1 - M_i^\top u_2)\} - 1]}$, $i = 1, \dots, l_+$. $M = [X_+, e_1]^\top \in \mathbb{R}^{(n+1) \times l_+}$, $N = [X_-, e_2]^\top \in \mathbb{R}^{(n+1) \times l_-}$; $u_1 = [u_+^\top, b_+]^\top \in \mathbb{R}^{n+1}$,

$u_2 = [u_-^\top, b_-^\top]^\top \in \mathbb{R}^{n+1}$. M_i^\top is the i^{th} row of M and N_j^\top is the j^{th} row of N . Further, for simplification, we use A_j and B_i to represent $(1 + N_j^\top u_1)$ and $(1 - M_i^\top u_2)$, respectively.

In accordance with the optimality condition, we obtain the following:

$$\nabla Q_1(u_1) = (MM^\top + C_1I)u_1 + \hat{N}s_1 = 0, \quad (11)$$

$$\nabla Q_2(u_2) = (NN^\top + C_3I)u_2 - \hat{M}s_2 = 0, \quad (12)$$

where $\hat{M} = [C_4M_1, \dots, C_4M_{l_+}]^\top \in \mathbb{R}^{(n+1) \times l_+}$, $\hat{N} = [C_2N_1, \dots, C_2N_{l_-}]^\top \in \mathbb{R}^{(n+1) \times l_-}$. $s_1 = [s_{11}, \dots, s_{1l_-}]^\top \in \mathbb{R}^{l_-}$, $s_2 = [s_{21}, \dots, s_{2l_+}]^\top \in \mathbb{R}^{l_+}$; $s_{1j} = \frac{\exp(aA_j)(aA_j+1)-1}{[1+A_j\{\exp(aA_j)-1\}]^2}$, $j = 1, \dots, l_-$; $s_{2i} = \frac{\exp(aB_i)(aB_i+1)-1}{[1+B_i\{\exp(aB_i)-1\}]^2}$, $i = 1, \dots, l_+$.

Now, we use Eqs. (11) and (12) to formulate iterative expressions for problems (9) and (10) in the following manner:

$$u_1^{t+1} = -(MM^\top + C_1I)^{-1} \hat{N}s_1^t, \quad (13)$$

$$u_2^{t+1} = (NN^\top + C_3I)^{-1} \hat{M}s_2^t. \quad (14)$$

Here, t represents the index of iteration. The iterative procedure involves iterating through Eqs. (13) and (14) until convergence is achieved. After obtaining the solutions, we can proceed to find the pair of hyperplanes (6).

To ascertain the class of a unseen sample $\tilde{x} \in \mathbb{R}^n$, we use the following decision rule:

$$\text{Class of } \tilde{x} = \arg \min_{i=+,-} \frac{|u_i^\top \tilde{x} + b_i|}{\|u_i\|}. \quad (15)$$

3.3 Non-linear GL-TSVM

For the non-linear case, we utilized the kernel trick to map the input data points to a higher dimensional space. The objective of non-linear GL-TSVM is to identify a pair of hypersurfaces of the following form:

$$g_+ = \kappa(x, X^\top) v_+ + b_+ = 0 \quad \text{and} \quad g_- = \kappa(x, X^\top) v_- + b_- = 0, \quad (16)$$

where $X^\top = [X_+; X_-]$ and κ is the kernel function.

To determine the hypersurfaces (16), we formulate the following optimization problems:
(Non-linear GL-TSVM-1)

$$\begin{aligned} \min_{v_+, b_+, \zeta_-} & \sum_{i=1}^{l_+} \frac{1}{2} \left(\kappa(x_i, X^\top) v_+ + b_+ \right)^2 + \frac{1}{2} C_1 \left(\|v_+\|_2^2 + b_+^2 \right) + C_2 \sum_{j=1}^{l_-} \zeta_j^-, \\ \text{s.t.} \quad & \zeta_j^- = 1 - \frac{1}{1 + (1 + \kappa(x_j, X^\top) v_+ + b_+) [\exp\{a(1 + \kappa(x_j, X^\top) v_+ + b_+)\} - 1]}, \\ & j = 1, \dots, l_-, \end{aligned} \quad (17)$$

(Non-linear GL-TSVM-2)

$$\begin{aligned}
& \min_{v_-, b_-, \zeta^+} \sum_{j=1}^{l_-} \frac{1}{2} \left(\kappa(x_j, X^\top) v_- + b_- \right)^2 + \frac{1}{2} C_3 \left(\|v_-\|_2^2 + b_-^2 \right) + C_4 \sum_{i=1}^{l_+} \zeta_i^+, \\
& \text{s.t.} \quad \zeta_i^+ = 1 - \frac{1}{1 + (1 - \kappa(x_i, X^\top) v_- - b_-) [\exp\{a(1 - \kappa(x_i, X^\top) v_- - b_-)\} - 1]}, \\
& \quad i = 1, \dots, l_+.
\end{aligned} \tag{18}$$

The iterative method to solve (17) and (18) can be derived as follows:

$$\begin{aligned}
v_1^{t+1} = & - \left(G G^\top + C_1 I \right)^{-1} \\
& \left(\sum_{j=1}^{l_-} C_2 H_j \frac{\exp\{a(1 + H_j^\top v_1^t)\} \{a(1 + H_j^\top v_1^t) + 1\} - 1}{[1 + (1 + H_j^\top v_1^t) \{\exp\{a(1 + H_j^\top v_1^t)\} - 1\}]^2} \right), \tag{19}
\end{aligned}$$

$$\begin{aligned}
v_2^{t+1} = & (H H^\top + C_3 I)^{-1} \\
& \left(\sum_{i=1}^{l_+} C_4 G_i \frac{\exp\{a(1 - G_i^\top v_2^t)\} \{a(1 - G_i^\top v_2^t) + 1\} - 1}{[1 + (1 - G_i^\top v_2^t) \{\exp\{a(1 - G_i^\top v_2^t)\} - 1\}]^2} \right). \tag{20}
\end{aligned}$$

Here $G = [\kappa(X_+, X^\top), e_1]^\top \in \mathbb{R}^{(l+1) \times l_+}$, $H = [\kappa(X_-, X^\top), e_2]^\top \in \mathbb{R}^{(l+1) \times l_-}$; G_i is the i^{th} column of the matrix G , H_j is the j^{th} column of the matrix H . $v_1 = [v_+^\top, b_+]^\top$, $v_2 = [v_-^\top, b_-]^\top$. Further, for simplification, we use E_j and F_i to represent $(1 + H_j^\top v_1)$ and $(1 - G_i^\top v_2)$, respectively.

It is important to highlight that Eqs. (19) and (20) involve complex matrix inversions. Therefore, to alleviate computational complexity, we utilized the Sherman Morrison Woodbury theorem [12]. Subsequently, in Eqs. (19) and (20), the inverse matrices are substituted with the following matrices:

$$P_1 = \frac{1}{C_1} \left(I - G (C_1 I + G^\top G)^{-1} G^\top \right), \tag{21}$$

$$P_2 = \frac{1}{C_3} \left(I - H (C_3 I + H^\top H)^{-1} H^\top \right). \tag{22}$$

Using the Eqs. (21) and (22), the iterative approach can be derived in the following manner:

$$v_1^{t+1} = -P_1 \hat{H} s_1^t, \tag{23}$$

$$v_2^{t+1} = P_2 \hat{G} s_2^t. \tag{24}$$

Here, $\hat{H} = [C_2 H_1, \dots, C_2 H_{l_-}] \in \mathbb{R}^{(l+1) \times l_-}$, and $\hat{G} = [C_4 G_1, \dots, C_4 G_{l_+}] \in \mathbb{R}^{(l+1) \times l_+}$. Additionally, $s_1^t \in \mathbb{R}^{l_-}$, $s_{1j}^t = \frac{\exp(a E_j^t) (a E_j^t + 1) - 1}{[1 + E_j^t \{\exp(a E_j^t) - 1\}]^2}$, $j = 1, \dots, l_-$;

$s_2^t \in \mathbb{R}^{l_+}$, $s_{2i}^t = \frac{\exp(aF_i^t)(aF_i^t+1)-1}{[1+F_i^t\{\exp(aF_i^t)-1\}]^2}$, $i = 1, \dots, l_+$. The iteration procedure is established by repeatedly applying Eqs. (23) and (24) until convergence is reached. Consequently, upon obtaining the solutions v_+, b_+ and v_-, b_- , we can then determine the positive and negative hypersurfaces generated by the kernel.

To predict the class of a new sample $\tilde{x} \in \mathbb{R}^n$, we use the following decision function:

$$\text{Class of } \tilde{x} = \arg \min_{i=+,-} \frac{|\kappa(\tilde{x}, X^\top) v_i + b_i|}{\sqrt{v_i^\top \kappa(X, X^\top) v_i}}. \quad (25)$$

The iterative algorithm structure for non-linear GL-TSVM subproblem (17) is clearly described in Algorithm 1. The structure for subproblem (18) is similar to it.

Algorithm 1. Non-linear GL-TSVM

Input:

Training dataset: $\{x_i, y_i\}_{i=1}^l$, $y_i \in \{-1, 1\}$;

The parameters: Convergence precision (η), maximum iteration number (T), parameter C_1 and C_2 , G-loss parameter a , iteration number $t = 0$;

Initialize: v_1^0 ;

Output: v_+, b_+ ;

1 : $G = [\kappa(X_+, X^\top), e_1]^\top$, $H = [\kappa(X_-, X^\top), e_2]^\top$.

2 : **while** $t \leq T$

3 : **for** $j \leftarrow 1$ to l_-

4 : $s_{1j}^t \leftarrow \frac{\exp(aE_j^t)(aE_j^t+1)-1}{[1+E_j^t\{\exp(aE_j^t)-1\}]^2}$

5 : **end for**

6 : $v_1^{t+1} \leftarrow -P_1 \hat{H} s_1^t$

7 : **if** $\|v_1^{t+1} - v_1^t\| < \eta$

8 : **break**

9 : **else**

10 : $t \leftarrow t + 1$

11 : **end if**

12 : **end while**

13 : $(v_+^\top, b_+)^T = v_1^{t+1}$.

3.4 Computational Complexity

Let l and n denote the number of samples and features in training dataset, respectively, and l_+ and l_- denote the count of positive and negative samples, respectively. The computational complexity of GL-TSVM primarily arises from the computation of matrix inversion. In the linear case, the algorithm requires solving the inverse of a matrix of order $(n+1) \times (n+1)$, which results in a time complexity of $\mathcal{O}((n+1)^3)$. Consequently, the computational complexity of the

proposed linear GL-TSVM is $\mathcal{O}(2T(n+1)^3)$, where T represents the maximum number of iterations. For the non-linear case, the algorithm needs to compute the inverse of two matrices: one of size $l_+ \times l_+$ and the other of size $l_- \times l_-$, with computational complexities of $\mathcal{O}(l_+^3)$ and $\mathcal{O}(l_-^3)$, respectively. Hence, for non-linear GL-TSVM, the computational complexity is $\mathcal{O}(T(l_+^3 + l_-^3))$. It is evident that non-linear GL-TSVM is not well-suited for large-scale problems due to its cubic computational complexity growth with respect to the size of positive and negative sample matrices. However, in future research, one can utilize the concept of granular computing to reduce the size of the sample matrices, making the non-linear GL-TSVM approach more feasible and effective for large-scale applications [16].

4 Experimental Evaluation

To validate the effectiveness of the proposed GL-TSVM model, we evaluate it on 25 UCI [7] and KEEL [6] benchmark datasets across various domains. For comparison, we used 6 state-of-the-art models, namely SVM [5], TSVM [11], Pin-GTSVM [23], SLTSVM [19], Wave-TSVM [3], and IF-RVFL [15]. Further, to showcase the efficacy of the proposed GL-TSVM in the biomedical realm, we evaluated it on the breast cancer (BreCaHis) and schizophrenia datasets. The detailed experimental setup employed for evaluating the models is provided in Section S.II of the supplementary file.

4.1 Evaluation on UCI and KEEL Datasets

For the linear case, the average classification accuracy of the proposed GL-TSVM and the baseline models are presented in Table 1. The detailed experimental results for each dataset are presented in Table S.I of the supplementary file. The average accuracies of the existing SVM, TSVM, Pin-GTSVM, SLTSVM, and Wave-TSVM are 83.67%, 86.99%, 87.33%, 84.18%, and 87.13%, respectively, whereas, the average accuracy of the proposed GL-TSVM is 87.52%, surpassing the compared models. In terms of average accuracy, the proposed GL-TSVM secured the top position, while the Pin-GTSVM achieved the second position with an accuracy difference of 0.19. Further, the average accuracy difference of proposed GL-TSVM from SVM, TSVM, SLTSVM, and Wave-TSVM are 3.85, 0.53, 3.34, and 0.39, respectively. This observation strongly underscores the competitiveness of the proposed linear GL-TSVM over baseline models. For the non-linear case, the average experimental results of the proposed GL-TSVM and baseline models are presented in Table 2. The detailed experimental results for each dataset can be found in Table S.II of the supplementary file. The average accuracies of SVM, TSVM, Pin-GTSVM, IF-RVFL, SLTSVM, Wave-TSVM, and the proposed GL-TSVM are 85.36%, 88.45%, 88.51%, 82.57%, 89.17%, 89.97%, and 90.52%, respectively. Evidently, the proposed GL-TSVM achieves the highest classification accuracy with an accuracy difference of 0.55 from the second-best model, Wave-TSVM. This finding firmly establishes the dominance of the non-linear GL-TSVM in comparison with the baseline models.

Table 1. Average accuracy and rank of linear GL-TSVM against baseline models on benchmark UCI and KEEL datasets.

	SVM [5]	TSVM [11]	Pin-GTSVM [23]	SLTSVM [19]	Wave-TSVM [3]	GL-TSVM [†]
Avg. Acc.	83.67	86.99	<u>87.33</u>	84.18	87.13	87.52
Avg. Rank	4	3.64	3.22	4.4	<u>2.88</u>	2.86

[†]represents the proposed model.

The boldface and underline indicate the best and second-best models, respectively.

Table 2. Average accuracy and rank of non-linear GL-TSVM against the baseline models on benchmark UCI and KEEL datasets.

	SVM [5]	TSVM [11]	Pin-GTSVM [23]	IF-RVFL [15]	SLTSVM [19]	Wave-TSVM [3]	GL-TSVM [†]
Avg. Acc.	85.36	88.45	88.51	82.57	89.17	<u>89.97</u>	90.52
Avg. Rank	5.26	4.08	3.9	5.92	3.68	<u>2.88</u>	2.28

[†]represents the proposed model.

The boldface and underline indicate the best and second-best models, respectively.

To further support the efficacy of the proposed GL-TSVM, we performed a thorough statistical analysis using the rank test, Friedman test, Nemenyi post hoc test, and win-tie-loss test. A detailed discussion of the statistical tests and their results is presented in Section S.III of the supplementary file.

4.2 Evaluation on Breast Cancer Dataset

To assess the effectiveness of the proposed GL-TSVM in practical scenarios, we evaluated it on breast cancer dataset named BreakHis, accessible at [20]. We utilized 1240 histopathology scans, each magnified 400 times. These scans predominantly fall into two categories: benign and malignant. Within the benign class, there are four subcategories: adenosis (ad), phyllodes tumor (pt), tubular adenoma (ta), and fibroadenoma (fd) with 106, 115, 130, and 237 scans, respectively. Similarly, the malignant class includes four subclasses: ductal carcinoma (dc), papillary carcinoma (pc), mucinous carcinoma (mc), and lobular carcinoma (lc) containing 208, 138, 169, and 137 scans, respectively. For feature extraction, we followed the same methodology as outlined in [10]. The average results of the proposed GL-TSVM compared to the baseline models on the BreakHis dataset are presented in Table 3. The detailed performance comparison on each dataset is outlined in Table S.III of the supplement file. The average accuracy of the proposed GL-TSVM stands at 75.67%, surpassing the baseline models SVM, TSVM, Pin-GTSVM, IF-RVFL, SLTSVM, and Wave-TSVM which achieved accuracies of 65.15%, 74.14%, 74.21%, 64.7%, 72.48%, and 74.2% respectively. Further, the average rank of the baseline models SVM, TSVM, Pin-GTSVM, IF-RVFL, SLTSVM, and Wave-TSVM is 6.22, 2.91, 3.19, 6.19, 4.38, and 3.09 while the average rank of GL-TSVM is 2.03, representing the most favorable position in comparison to the baseline models. Further, the average rank differences between the proposed GL-TSVM and the other baseline models

Table 3. Average performance of the proposed GL-TSVM against the baseline models on the BreakHis dataset.

	SVM [5]	TSVM [11]	Pin-GTSVM [23]	IF-RVFL [15]	SLTSVM [19]	Wave-TSVM [3]	GL-TSVM [†]
Avg. Acc.	65.15	74.14	<u>74.21</u>	64.7	72.48	74.2	75.67
Avg. Rank	6.22	<u>2.91</u>	3.19	6.19	4.38	3.09	2.03

[†]represents the proposed model.

The boldface and underline indicate the best and second-best models, respectively.

Table 4. Average performance of the proposed GL-TSVM against the baseline models on the schizophrenia dataset.

	SVM [5]	TSVM [11]	Pin-GTSVM [23]	IF-RVFL [15]	SLTSVM [19]	Wave-TSVM [3]	GL-TSVM [†]
Acc.	67.57	75	<u>77.38</u>	74.33	78.39	75	78.39

[†]represents the proposed model.

The boldface and underline indicate the best and second-best models, respectively.

(SVM, TSVM, Pin-GTSVM, IF-RVFL, SLTSVM, and Wave-TSVM) are 4.19, 0.88, 1.46, 4.16, 2.35 and 1.06, respectively, showing substantial performance advantages for GL-TSVM over these models. Overall, the results showcase that the proposed GL-TSVM is significantly superior in the domain of breast cancer diagnosis when compared to the baseline models.

4.3 Evaluation on Schizophrenia Dataset

To further demonstrate the competitiveness of the proposed GL-TSVM model, we evaluated it for diagnosing schizophrenia patients. The data used in this study was obtained from the center for biomedical research excellence (COBRE) (http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html). The dataset includes 72 schizophrenia subjects (ages 18–65, mean age 38.1 ± 13.9 years) and 74 healthy control subjects (ages 18–65, mean age 35.8 ± 11.5 years). The feature extraction process followed the methodology outlined in [24]. Table 4 presents a comparative analysis of the performance of the proposed GL-TSVM model against baseline models. SVM has the lowest accuracy at 67.57%. TSVM, Pin-GTSVM, and Wave-TSVM show improved accuracies of 75%, 77.38%, and 75%, respectively, while IF-RVFL achieves 74.33%. Notably, the SLTSVM model excels with an accuracy of 78.39%, matched by the proposed GL-TSVM model. This indicates that the proposed GL-TSVM is as effective as the best-performing models, validating its efficacy as a reliable and competitive tool for schizophrenia diagnosis.

4.4 Effectiveness of the G-Loss Function

The experimental findings affirm the efficacy of the G-loss function in bolstering the robustness and performance of GL-TSVM. The asymmetric design of the G-loss function permits differential handling of samples relative to their distance from the decision boundary, effectively reducing the influence of noise. Its

bounded characteristic caps the maximum loss, thereby preventing extreme values from unduly affecting the model. Additionally, the smoothness of the G-loss function supports the implementation of gradient-based optimization techniques, facilitating both efficient and stable convergence. Collectively, these features significantly enhance the performance of the proposed GL-TSVM, as demonstrated by the experimental results across a diverse array of datasets and domains.

5 Conclusions

In conclusion, the introduction of the G-loss function and the development of GL-TSVM model have significantly enhanced the robustness and performance of traditional TSVM algorithms. By leveraging the asymmetry, boundedness, and smoothness properties of the G-loss function, the GL-TSVM model offers a novel approach to handling outliers or noise in the data. Furthermore, the inclusion of a regularization term to adhere to the structural risk minimization principle led to the creation of a more powerful classifier in the form of GL-TSVM. The iterative algorithm utilized for optimizing the GL-TSVM model ensures efficient convergence and stability. The experimental evaluations on a diverse set of benchmark datasets have consistently demonstrated the superior performance of GL-TSVM compared to baseline models. The application of GL-TSVM to breast cancer (BreCaKHis) and schizophrenia datasets further validates its effectiveness in the biomedical domain.

However, it is noteworthy that, due to the computation of matrix inversion, GL-TSVM is not well-suited for large-scale problems. In the future, one can reformulate the GL-TSVM to circumvent the need to compute matrix inversion. Further, in the future, researchers can explore the fusion of the G-loss function with cutting-edge methodologies like support matrix machines [13] to tackle complex real-world problems.

Acknowledgment. This project receives support from the Science and Engineering Research Board through the Mathematical Research Impact-Centric Support (MATRICS) scheme, with Grant No. MTR/2021/000787. Additionally, Mushir Akhtar's research fellowship is provided by the Council of Scientific and Industrial Research (CSIR), New Delhi, under Grant No. 09/1022(13849)/2022-EMR-I.

References

1. Akhtar, M., Tanveer, M., Arshad, M.: RoBoSS: a robust, bounded, sparse, and smooth loss function for supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* (2024). <https://doi.org/10.1109/TPAMI.2024.3465535>
2. Akhtar, M., Tanveer, M., Arshad, M.: HawkEye: advancing robust regression with bounded, smooth, and insensitive loss function. *arXiv preprint arXiv:2401.16785*
3. Akhtar, M., Tanveer, M., Arshad, M., and for the Alzheimer's Disease Neuroimaging Initiative: Advancing supervised learning with the wave loss function: a robust and smooth approach. *Pattern Recognit.*, 110637 (2024). <https://doi.org/10.1016/j.patcog.2024.110637>

4. Borah, P., Gupta, D.: Functional iterative approaches for solving support vector classification problems based on generalized Huber loss. *Neural Comput. Appl.* **32**(13), 9245–9265 (2020)
5. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
6. Derrac, J., Garcia, S., Sanchez, L., Herrera, F.: KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *J. Mult. Valued Logic Soft Comput.* **17**, 255–287 (2015)
7. Dua, D., Graff, C.: UCI machine learning repository **7**(1), 62 (2017). <http://archive.ics.uci.edu/ml>
8. Ganaie, M.A., Tanveer, M., for the Alzheimer's Disease Neuroimaging Initiative: KNN weighted reduced universum twin SVM for class imbalance learning. *Knowl.-Based Syst.* **245**, 108578 (2022)
9. Ganaie, M.A., Tanveer, M., Lin, C.T.: Large-scale fuzzy least squares twin SVMs for class imbalance learning. *IEEE Trans. Fuzzy Syst.* **30**(11), 4815–4827 (2022). <https://doi.org/10.1109/TFUZZ.2022.3161729>
10. Gautam, C., et al.: Minimum variance-embedded deep kernel regularized least squares method for one-class classification and its applications to biomedical data. *Neural Netw.* **123**, 191–216 (2020)
11. Jayadeva, Khemchandani, R., Chandra, S.: Twin support vector machines for pattern classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(5), 905–910 (2007)
12. Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. *Expert Syst. Appl.* **36**(4), 7535–7543 (2009)
13. Kumari, A., Akhtar, M., Shah, R., Tanveer, M.: Support matrix machine: a review. *arXiv preprint arXiv:2310.19717* (2023)
14. Kumari, A., Akhtar, M., Tanveer, M., Arshad, M.: Diagnosis of breast cancer using flexible pinball loss support vector machine. *Appl. Soft Comput.*, 111454 (2024). <https://doi.org/10.1016/j.asoc.2024.111454>
15. Malik, A.K., Ganaie, M.A., Tanveer, M., Suganthan, P.N., for the Alzheimer's Disease Neuroimaging Initiative: Alzheimer's disease diagnosis via intuitionistic fuzzy random vector functional link network. *IEEE Trans. Comput. Soc. Syst.*, 1–12 (2022). <https://doi.org/10.1109/TCSS.2022.3146974>
16. Quadir, A., Tanveer, M.: Granular ball twin support vector machine with pinball loss function. *IEEE Trans. Comput. Soc. Syst.*, 1–10 (2024). <https://doi.org/10.1109/TCSS.2024.3411395>
17. Quadir, A., Akhtar, M., Tanveer, M.: Enhancing multiview synergy: robust learning by exploiting the wave loss function with consensus and complementarity principles. *arXiv preprint arXiv:2408.06819* (2024)
18. Shao, Y., Zhang, C., Wang, X., Deng, N.: Improvements on twin support vector machines. *IEEE Trans. Neural Networks* **22**(6), 962–968 (2011)
19. Si, Q., Yang, Z., Ye, J.: Symmetric LINEX loss twin support vector machine for robust classification and its fast iterative algorithm. *Neural Netw.* **168**, 143–160 (2023)
20. Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L.: A dataset for breast cancer histopathological image classification. *IEEE Trans. Biomed. Eng.* **63**(7), 1455–1462 (2015)
21. Tanveer, M.: Robust and sparse linear programming twin support vector machines. *Cogn. Comput.* **7**(1), 137–149 (2015)
22. Tanveer, M., Gautam, C., Suganthan, P.N.: Comprehensive evaluation of twin SVM based classifiers on UCI datasets. *Appl. Soft Comput.* **83**, 105617 (2019). <https://doi.org/10.1016/j.asoc.2019.105617>

23. Tanveer, M., Sharma, A., Suganthan, P.N.: General twin support vector machine with pinball loss function. *Inf. Sci.* **494**, 311–327 (2019)
24. Tanveer, M., Ganaie, M.A., Bhattacharjee, A., Lin, C.T.: Intuitionistic fuzzy weighted least squares twin SVMs. *IEEE Trans. Cybern.* **53**(7), 4400–4409 (2022)
25. Tanveer, M., Rajani, T., Rastogi, R., Shao, Y., Ganaie, M.A.: Comprehensive review on twin support vector machines. *Ann. Oper. Res.*, 1–46 (2022). <https://doi.org/10.1007/s10479-022-04575-w>
26. Tanveer, M., Tiwari, A., Choudhary, R., Ganaie, M.A.: Large-scale pinball twin support vector machines. *Mach. Learn.* (13), 1–24 (2021). <https://doi.org/10.1007/s10994-021-06061-z>
27. Wang, Q., Ma, Y., Zhao, K., Tian, Y.: A comprehensive survey of loss functions in machine learning. *Ann. Data Sci.* **9**, 1–26 (2020)
28. Zheng, X., Zhang, L., Yan, L.: CTSVM: a robust twin support vector machine with correntropy-induced loss function for binary classification problems. *Inf. Sci.* **559**, 22–45 (2021)



An Approach Towards Learning K-Means-Friendly Deep Latent Representation

Debapriya Roy^(✉) 

Institute of Engineering and Management, Kolkata, India
debapriyakundu1@gmail.com

Abstract. Clustering is a long-standing problem area in data mining. The centroid-based classical approaches to clustering mainly face difficulty in the case of high dimensional inputs such as images. With the advent of deep neural networks, a common approach to this problem is to map the data to some latent space of comparatively lower dimensions and then do the clustering in that space. Network architectures adopted for this are generally autoencoders that reconstruct a given input in the output. To keep the input in some compact form, the encoder in AE's learns to extract useful features that get decoded at the reconstruction end. A well-known centroid-based clustering algorithm is K-means. In the context of deep feature learning, recent works have empirically shown the importance of learning the representations and the cluster centroids together. However, in this aspect of joint learning, recently a continuous variant of K-means has been proposed; where the softmax function is used in place of argmax to learn the clustering and network parameters jointly using stochastic gradient descent (SGD). However, unlike K-means, where the input space stays constant, here the learning of the centroid is done in parallel to the learning of the latent space for every batch of data. Such batch updates disagree with the concept of classical K-means, where the clustering space remains constant as it is the input space itself. To this end, we propose to alternatively learn a clustering-friendly data representation and K-means based cluster centers. Experiments on some benchmark datasets have shown improvements of our approach over the previous approaches.

1 Introduction

Clustering is a method of finding the inherent pattern in data by segregating it into different groups. Primarily, it is used to partition unlabeled data into groups for extracting meaningful information. It has various applications in recommender systems where user queries are often grouped to give informed product suggestions, resulting in a better purchasing pattern. In information retrieval systems, this is also used for partitioning similar or associated articles into the same clusters to enhance the efficiency and effectiveness of the retrieval process.

In addition, clustering has immense applicability in image segmentation, medical imaging, social network analysis, anomaly detection, market segmentation, etc.

The idea of grouping similar items needs some distance metric. As the complex manifold in high-dimensional input space makes the use of Euclidean distance less meaningful, therefore, classical clustering methods like K-means and GMM are less effective there. This brings up the idea of clustering in latent space, a comparatively low-dimensional space. However, learning in such a space is often challenging as we are dealing with an unsupervised problem. Employing autoencoders (AE) in learning the “clustering friendly” latent space has facilitated the development of deep clustering approaches over the last few years [2, 5, 12].

Based on the ways to learn an embedding space with latent representations suitable for clustering, the existing approaches to deep clustering can be primarily categorized into three types. The first type of method [11] learns the latent representations first in the pretraining phase, then optimizes the representations for clustering using some clustering loss while learning the clustering parameters. This type of method finetunes the embedding space for clustering without regard to its reconstructability. The very first work in this direction is Deep Embedded Clustering (DEC) [11]. DEC employs a clustering loss to finetune the encoder of the pretrained AE for clustering while learning the cluster centers. Initially, the soft assignments between the embedded points and the cluster centers are computed. The KL divergence-based clustering loss is employed to improve upon this initial soft estimate by learning from high-confidence predictions. This is done by pushing the soft estimates toward the hard estimates. As DEC abandons the decoder and finetunes the encoder using only the clustering loss, this might distort the embedded space, causing a loss of representativeness of the data. But employing the clustering loss only for finetuning could distort the embedded space to the extent of weakening the representativeness of the latent features, which in turn could hurt the clustering performance.

Considering this, the second type of method [5, 12] proposed a joint autoencoder (AE)-based dimensionality reduction (DR) and K-means objective. In IDEC, the DR and cluster center learning are done jointly. Whereas the target distribution is updated every T iterations of DR and cluster center updates. Unlike DEC and IDEC, where the clustering loss is based on KL-divergence, DCN [12] adopted a clustering loss related to the classical K-means. However, in K-means, the membership values are restricted to a discrete set, causing the joint optimization of clustering and the reconstruction loss to be numerically infeasible. DCN deals with this by jointly optimizing the reconstruction and the clustering loss in alternating stochastic optimization, where gradient update and discrete cluster assignments are done alternatively. In DCN, this is optimized using alternative stochastic optimization; which implies the main objective function is divided into two different objectives that are optimized alternatively. Precisely, the DR part is optimized while keeping the K-means parameters constant, followed by K-means optimization while keeping the DR part constant. In the third type of method [2], in place of K-means, deep K-means is used, which is a continuous variant of K-means as mentioned previously. However, to

learn the cluster centers and the data representations jointly, DKM [2] proposed a continuous variant of K-means, where the argmax of K-means is replaced with one of its soft variants, which is the softmax function. Using deep K-means in place of classical K-means allows to replace the discrete optimization steps with joint optimization AE’s parameters and clustering parameters using SGD.

As DKM employs SGD, clustering and network parameter updates are done for every batch of data. However, in classical K-means, the input space remains constant, which is not the case in batch updates as in DKM. To address this, we propose to separately update the network parameters and clustering parameters. The network parameters are updated by jointly optimizing the reconstruction loss and our proposed CenTering (CT) loss function. The CT loss pushes the latent space into being suitable for clustering. Clustering parameters, i.e., the centroids, are then learned on the latent space by optimizing the classical K-means objective. Therefore, the network and the clustering parameter updates are done alternatively for every training epoch. Experiments on some benchmark datasets show that our method can achieve a better Normalized Mutual Information (NMI) and ACCuracy (ACC) score comparatively. In the rest of the paper, we present a brief literature survey in Sect. 2 and our methodology in Sect. 3. We present qualitative and quantitative experiments with an ablation study in Sect. 4. Finally, we conclude in Sect. 5.

Notations: Before going further, let us introduce the notations. We consider a dataset of N points, $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^m : i = 1, \dots, N\}$, where m is the dimension of the input. In the case of images of shapes $(h \times w \times c)$, the data is flattened to the dimension hwc , where h, w, c refers to the height, width, and number of channels of an image. The goal is to cluster \mathbf{X} into K clusters, which is similar in concept to the classes in supervised settings. The set of the centroids of the K clusters is represented by $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k\}$. Here \mathbf{r}_k is the centroid or representative of the cluster k . In autoencoder we denote the encoder by $h_\theta : \mathbb{R}^m \rightarrow \mathbb{R}^l$, where l is the dimension of latent space or embedding space and θ is the parameters of the encoder. The decoder is represented by the mapping $h_\phi : \mathbb{R}^l \rightarrow \mathbb{R}^m$, where ϕ denotes the parameters of the decoder. We use bold font for vectors. Note that throughout the paper we have used the terms centers and centroids alternatively, while both denote the same thing, i.e., the cluster centers.

2 Related Works

The problem of clustering has been well studied over the years. However, studies on leveraging the features learned by deep neural networks for clustering have come to light over the past few years [2–6, 8, 9, 11–13] etc.

In this section, we discuss developments in K-means employing deep neural networks. We start by discussing briefly the formulation of the classical K-means [1] algorithm, followed by a brief discussion on the autoencoder (AE), which is used to learn the latent features of the data. Following this, we discuss the formulations of related previous approaches in the context of the current problem.

2.1 Classical K-Means Algorithm

Let \mathbf{x} denotes an object from a set $\{\mathbf{X} = \mathbf{x}_i \in \mathbb{R}^m : i = 1, \dots, N\}$ of objects to be clustered. The task of clustering is to group N data samples into K clusters. The K-means algorithm attempts this task by optimizing the following objective function,

$$\min_{\mathbf{R}} \sum_{i=1}^N \|\mathbf{x}_i - c(\mathbf{x}_i; \mathbf{R})\|_2^2, \quad (1)$$

$$\text{with } c(\mathbf{x}_i; \mathbf{R}) = \arg \min_{\mathbf{r} \in \mathbf{R}} \|\mathbf{x}_i - \mathbf{r}\|_2^2,$$

where $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k\}$. Here \mathbf{r}_k is the representative of the cluster k and \mathbf{R} is the set of all the representatives. $c(\mathbf{x}_i; \mathbf{R})$ gives the closest representative of \mathbf{x}_i in terms of L2 distance.

2.2 Autoencoder

Among the multiple types of deep neural networks, an autoencoder is a self-supervised deep learner that is trained using an identity function $\mathbf{X} = F_{\theta, \phi}(\mathbf{X}) = g_{\phi}(h_{\theta}(\mathbf{X}))$, where F is the learnable function or the autoencoder as a whole and h_{θ} and g_{ϕ} are respectively the encoder and the decoder function, which represents the mapping from the data space to the latent or encoding space and the reverse mapping from the encoding space back to the data space, respectively. In general, the objective of AE is,

$$\min_{\theta, \phi} \sum_{\mathbf{x} \in \mathbf{X}} \|F_{\theta, \phi}(\mathbf{x}) - \mathbf{x}\|_2^2. \quad (2)$$

In general, AE is used for dimensionality reduction (DR) and noise reduction. In problems related to clustering, AE is mostly used for DR to tackle the issue of the curse of dimensionality, which often causes the data space to be unsuitable for clustering.

2.3 Autoencoder (AE)-Based Deep Clustering Variants

Optimizing the latent space of an AE for clustering was first proposed in DEC [11]. In DEC, Xie et al. [11] proposed an approach that jointly optimizes the centroids \mathbf{R} and the encoder's parameters θ using Stochastic Gradient Descent (SGD). DEC in particular solves the following objective,

$$L = \min_{\theta, \mathbf{R}} \text{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \quad (3)$$

where p_{ij} is a function of q_{ij} as the following,

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_{j'} (q_{ij'}^2 / \sum_i q_{ij'})} \quad (4)$$

$$q_{ij} = \frac{(1 + \|f_{\theta}(\mathbf{x}_i) - \mathbf{r}_j\|_2^2 / \alpha^t)^{-\frac{\alpha^t+1}{2}}}{\sum_{j'} (1 + \|f_{\theta}(\mathbf{x}_i) - \mathbf{r}_{j'}\|_2^2 / \alpha^t)^{-\frac{\alpha^t+1}{2}}}, \quad (5)$$

q_{ij} is the similarity between the embedded point $z_i = f_{\theta}(\mathbf{x}_i)$ and the j^{th} cluster centroid \mathbf{r}_j , interpreted as the probability of assigning sample i to cluster j . This is measured using Student's t-distribution as a kernel with α^t as the degrees of freedom.

In IDEC, Guo et al. [5] claimed that employing only the clustering loss might not preserve the local structure of the data in the latent space, leading to a corrupted latent space. Therefore, in [12] the objective is revised to the following,

$$\min_{\theta, \phi, \mathbf{R}} \sum_{i=1}^N \|F_{\theta, \phi}(\mathbf{x}_i) - \mathbf{x}_i\|_2^2 + \lambda_{idec} \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \quad (6)$$

where $\lambda_{idec} > 0$ is a coefficient to control the degree of distortion of the latent space for the goal of clustering. A similar idea is proposed by Yang et al. [12], but instead of divergence, they used a L2 norm based distance measure. Their objective function is the following,

$$\min_{\theta, \phi, \mathbf{R}} \sum_{i=1}^N \|F_{\theta, \phi}(\mathbf{x}_i) - \mathbf{x}_i\|_2^2 + \frac{\lambda_{dcn}}{2} \|h_{\theta}(\mathbf{x}_i) - \mathbf{M}\mathbf{s}_i\|_2^2, \quad (7)$$

where \mathbf{M} is the matrix with its k^{th} column being the k^{th} centroid \mathbf{r}_k and \mathbf{s}_i is the assignment vector of data point i , having only 1 non-zero value in the position of assigned cluster k . $\lambda_{dcn} \geq 0$ is a regularization parameter having a similar role as λ_{idec} . SGD can not be directly applied to jointly optimize $\theta, \phi, \mathbf{M}, \mathbf{S}_i$ together as \mathbf{s}_i is constrained on a discrete set. Therefore, (θ, ϕ) and $(\mathbf{M}, \mathbf{S}_i)$ are optimized in alternating optimization.

To pursue joint optimization DKM [2] revised the above objective in the following way,

$$\min_{\theta, \phi, \mathbf{r}_j} \sum_{i=1}^N \|F_{\theta, \phi}(\mathbf{x}_i) - \mathbf{x}_i\|_2^2 + \lambda_{dkm} \sum_{k=1}^k \|h_{\theta}(\mathbf{x}_i) - \mathbf{r}_k\|_2^2 G_k^{dkm}(h_{\theta}(\mathbf{x}_i), \alpha_{dkm}; \mathbf{R}), \quad (8)$$

where, $G_k^{dkm}(\cdot)$ is defined as follows,

$$G_k^{dkm}(h_{\theta}(\mathbf{x}_i), \alpha_{dkm}; \mathbf{R}) = \frac{e^{-\alpha_{dkm} \|h_{\theta}(\mathbf{x}_i) - \mathbf{r}_k\|_2^2}}{\sum_{k'=1}^K e^{-\alpha_{dkm} \|h_{\theta}(\mathbf{x}_i) - \mathbf{r}_{k'}\|_2^2}} \quad (9)$$

From Eq. 7 to Eq. 8 notice the change is only in the second term where G_k^{dkm} is introduced.

Compared to DKM, our formulation differs in two ways, first, our formulation of $G_{k,f}$ is different; second, we reinitialize the cluster centers after every epoch by applying K-means to the feature space data. IDEC and DCN completely separate

feature learning and clustering; instead, we learn clustering-friendly features and then do the clustering alternatively until a convergence criterion based on a loss function is met. Thus, the latent space or feature space is always being learned based on some clustering criterion.

3 Proposed Formulation

Given a dataset \mathbf{X} having N points $\{\mathbf{x}_i \in \mathbb{R}^m : i = 1, \dots, N\}$ and K clusters, our goal is to assign each point to one of the K clusters. We attempt to solve this problem in two steps. In the first step, we learn a feature space that reduces the dimensionality of the data while learning a suitable clustering embedding. For this, we finetune a pretrained autoencoder using two loss functions. First, the reconstruction loss is used to maintain the representability of the data while clustering. Second, our proposed centering (CT) loss minimizes the weighted distance between the cluster centers and the data embeddings. In the second step, we simply optimize the objective of the classical K-means on the data embeddings obtained from the encoder to reinitialize the k cluster centers. Using the above reasoning, we came up with the following objective function,

$$\min_{\theta, \phi, \mathbf{R}} \underbrace{\sum_{i=1}^N (l(g_\phi(h_\theta(\mathbf{x}_i)), \mathbf{x}_i))}_{\text{reconstruction loss}} + \lambda \underbrace{\sum_{k=1}^K \|h_\theta(\mathbf{x}_i) - \mathbf{r}_k\|_2^2 G_{K,f}(h_\theta(\mathbf{x}_i), \alpha; \mathbf{R}))}_{\text{centering loss}}, \quad (10)$$

where,

$$G_{K,f}(h_\theta(\mathbf{x}_i), \alpha; \mathbf{R}) = \frac{\frac{1}{\bar{f}(h_\theta(\mathbf{x}_i), \mathbf{r}_k)^\alpha}}{\sum_{k'=1}^K \frac{1}{\bar{f}(h_\theta(\mathbf{x}_i), \mathbf{r}_{k'})^\alpha}}, \quad (11)$$

where $f(\cdot, \cdot) = \|\cdot - \cdot\|_2^2$. Here, $G_{K,f}(\cdot, \cdot)$ is a differentiable function with respect to θ, \mathbf{R} . $\alpha \in \mathbb{R}^+$ is a parameter. In general we have observed that $\alpha \geq 2$ gives better clustering performance. Equation 10 is optimized using SGD.

Since our goal is to learn some K-means friendly data representation, therefore, at the end of every epoch we compute K-means on the latent space to initialize the centers \mathbf{R} for the next epoch. Therefore, the values of \mathbf{R} are only used to learn the latent representation of the data during the SGD which is suitable for clustering.

The steps of our approach are summarized in the Algorithm 1.

4 Experiments

In this section, we conduct qualitative and quantitative experimental analyses. We compare with some benchmark methods on two standard clustering metrics ACCuracy (ACC) and Normalized Mutual Information (NMI) which are discussed below.

Algorithm 1. The proposed method**Input:** $A_1 \dots A_N$ **Output:** Sum (sum of values in the array)

- 1: Initialise θ, ϕ, \mathbf{R} randomly.
- 2: Pretrain the AE for n_p number of pretraining epochs.
- 3: Initialise the centers \mathbf{R} using K-means of the latent representations of the dataset,
 $L_{\mathbf{X}} = \{h_{\theta}(\mathbf{x}_i), i = 1, \dots, N\}$
- 4: **for** $epoch \leftarrow 1$ to n_e number of epochs **do**
- 5: **for** $batch \leftarrow 1$ to n_b number of batches **do**
- 6: Sample minibatch of m samples $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ from the dataset
- 7: Optimize the objective function in Eq. 10
- 8: **end for**
- 9: Compute the latent representations of the dataset, $L_{\mathbf{X}} = \{h_{\theta}(\mathbf{x}_i), i = 1, \dots, N\}$
- 10: Optimize the K-means objective function on $L_{\mathbf{X}}$ for finding the centroids \mathbf{R} of
the K clusters.
- 11: **end for**
- 12: **return**

4.1 Datasets

We conduct experiments on MNIST, USPS, COIL100, CMU-PIE and RCV1-v2 datasets. RCV1-v2 is text dataset and the rest are image datasets. The details of different datasets are given in Table 1.

Table 1. .

Dataset	MNIST	USPS	COIL100	CMU-PIE	RCV1-v2
#Samples	70000	11000	7200	2856	10,000
#Catagories	10	10	100	68	4
Image size	28×28	16×16	128×128	32×32	–
Input dimension	784	256	16384	1024	2000

4.2 Evaluation Metrics

Accuracy: Considering c_i as the cluster assignment of data x_i and y_i as its ground truth cluster label, the *accuracy* (ACC) of a clustering model is defined as,

$$ACC = \max_m \frac{\sum_{i=1}^N \mathbf{1}\{y_i == m(c_i)\}}{N}, \quad (12)$$

where m ranges over all possible one-to-one mappings between clusters and labels. N is the total number of data samples in the dataset. Intuitively, this metric finds the best match between the algorithm’s cluster assignment and the ground truth assignment. In general, the Hungarian algorithm [7] is best to compute this mapping.

Normalized Mutual Information (NMI): It is an information theory based similarity measure in clustering. It is bounded in $[0, 1]$ and equates to 1 when the ground truth and the predicted clustering are equal. Considering C as the cluster distribution, Y as the ground truth, and $H(\cdot)$ as the entropy, the NMI value is computed as,

$$NMI = \frac{2H(C, Y)}{H(C) + H(Y)} \quad (13)$$

NMI approaches 1 when distribution C is similar to Y and in the opposite case NMI approaches 0.

We compare the ACC and NMI scores with that of others in Tables 2, 3.

4.3 Algorithms in Comparison

- KM [1]: The classical K-means clustering approach.
- AEKM: Here, before applying K-means, the dimensionality of the data is first reduced using an autoencoder (AE). Therefore, the clustering is done in the latent space or embedding space learned by the autoencoder.
- DCN [12]: It proposes a joint dimensionality reduction and clustering approach to recover a ‘clustering friendly’ latent representation. It employs alternating stochastic optimization to update the clustering parameters (i.e., the cluster centroids) and the network parameters alternatively. Here, the parameters of the AE are initialized by pre-training before employing clustering.
- DKM [2] - Similar to DCN, it also proposes an autoencoder based approach for clustering. However, unlike DCN, it proposes a continuous variant of the K-means objective function to jointly achieve dimensionality reduction and clustering using gradient descent. The proposed objective function is fully differentiable with respect to both the clustering and the network parameters. Here also, the AE network is pretrained before adding the clustering loss to the objective.

4.4 Experimental Settings

For every method, we report the average score over 10 runs with different seeds. Note that, same set of 10 seeds is taken for every method to maintain fairness in comparison. For the methods requiring pretraining of the AE model, i.e., DCN, DKM, and ours, we have pretrained for 50 epochs. The finetuning is done for 100 epochs. A fixed batch size of 256 is used for all the experiments. The optimal parameter values for all the parametric methods, i.e., DCN, DKM, and ours, are obtained by grid-search over the feasible set of parameter values. The optimal parameter values are reported in Table 4.

4.5 Quantitative Analysis

The ACC and NMI scores reported in Table 2 and Table 3 show that our method achieves better scores in terms of both metrics. Table 4 shows the values of λ , the

Table 2. ACC values over different datasets.

Dataset	MNIST	USPS	COIL100	CMU-PIE	RCV1
KM	53.50 ± 0.30	67.3 ± 0.10	49.51 ± 1.13	21.31 ± 0.68	50.8 ± 2.90
AEKM	80.80 ± 1.80	72.9 ± 0.80	49.66 ± 0.84	24.17 ± 1.54	56.70 ± 3.60
DCN	81.10 ± 1.90	73.0 ± 0.80	49.23 ± 0.88	24.78 ± 1.66	56.70 ± 3.60
DKM	84.00 ± 2.20	75.7 ± 1.30	49.50 ± 0.78	31.61 ± 0.86	58.3 ± 3.80
Ours	90.01 ± 5.83	78.74 ± 4.21	51.56 ± 0.94	32.40 ± 1.81	60.39 ± 2.33

Table 3. NMI values over different datasets.

Dataset	MNIST	USPS	COIL100	CMU-PIE	RCV1
KM	49.8 ± 0.5	61.4 ± 0.1	76.82 ± 0.35	41.67 ± 0.67	31.3 ± 5.4
AEKM	75.2 ± 1.1	16.9 ± 1.3	77.03 ± 0.30	51.43 ± 1.87	31.5 ± 4.3
DCN	75.7 ± 1.1	71.9 ± 1.2	76.76 ± 0.47	52.33 ± 1.88	31.6 ± 4.3
DKM	79.6 ± 0.9	77.6 ± 1.1	77.82 ± 0.25	61.92 ± 0.82	33.1 ± 4.9
Ours	87.64 ± 1.55	80.64 ± 1.20	78.17 ± 0.40	62.18 ± 0.89	35.85 ± 3.03

coefficient of the clustering loss. λ keeps the balance between the reconstruction and the clustering loss to achieve optimal clustering performance. We see that for the CMU-PIE and RCV1-v2 datasets, the λ value is low, indicating that the deep embeddings from the pretrained model are relatively good for clustering. Whereas for the MNIST, USPS, and COIL100 datasets, the coefficient of the clustering loss is quite large comparatively, indicating the importance of the clustering loss along with the centroid reinitialization approach for improved clustering outcomes. An important observable in Table 4 is that the λ values of ours on the MNIST, USPS, and COIL100 datasets are higher compared to that of DKM, which shows that our approach plays a more significant role in learning clustering-related features compared to that of DKM in these datasets.

Table 4. Optimal values of parameters of different methods.

Dataset	Pretraining epochs	Fine-tuning epochs	Batch size	α	λ_{Ours}	λ_{DKM}	λ_{DCN}
MNIST	50	100	256	3	1e+1	1e+0	1e+1
USPS	50	100	256	2.5	1e+1	1e+0	1e-1
COIL100	50	100	256	3	1e+1	1e-1	1e+0
CMU-PIE	50	100	256	2	1e-3	1e-2	1e-1
RCV1	50	100	256	2	1e-4	1e-2	1e-1

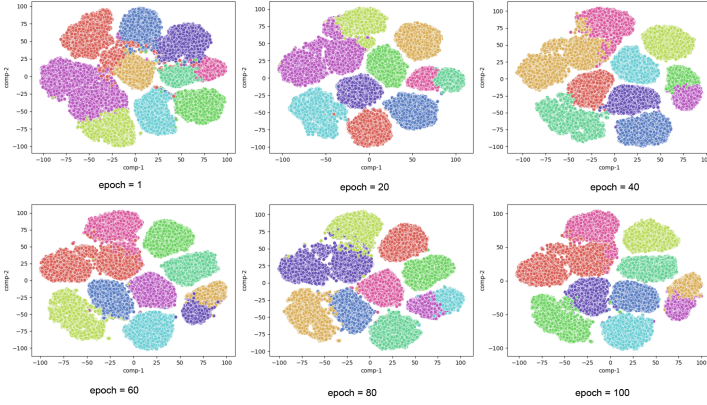


Fig. 1. The visualization of the learned clusters in the latent space of the full MNIST dataset over the different fine-tuning epochs of our method.

4.6 Qualitative Analysis

To compare with DKM qualitatively, we have given the TSNE [10] plot of deep embeddings of the full-MNIST dataset with the different predicted cluster labels of our method and that of DKM in Fig. 1, 2, respectively. In order to analyze the clustering progression, we provided the plot over the different fine-tuning epochs. For fair comparison, both methods have been executed under the same experimental conditions. It can be observed that from the first fine-tuning epoch, our method shows better cluster compactness that improves as the epochs increase. Finally, we see that our clustering shows increased inter-cluster distance compared to that of DKM, resulting in better clustering metric values; that can be verified from the quantitative analysis.

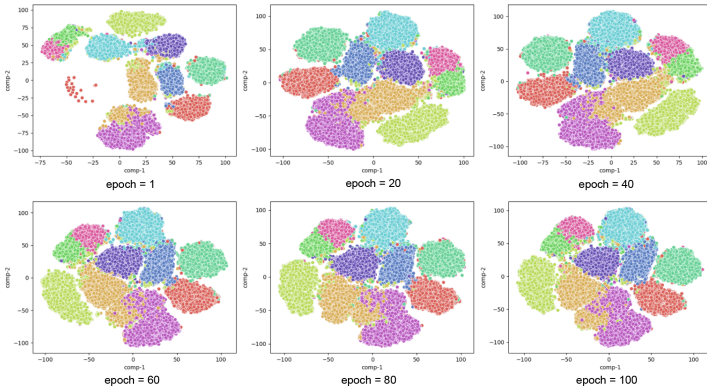


Fig. 2. The visualization of the learned clusters in the latent space of the full MNIST dataset over different fine-tuning epochs of DKM.

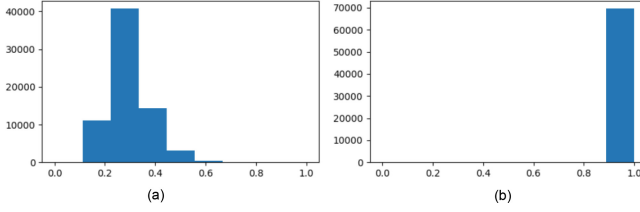


Fig. 3. Histograms of the highest cluster membership value of each data point on MNIST, (a) before finetuning, (b) after finetuning at the 100th epoch. Observe that before finetuning the cluster memberships are in the low confidence region, indicating poor clustering performance. Whereas after finetuning all the memberships are in high confidence, indicating better clustering.

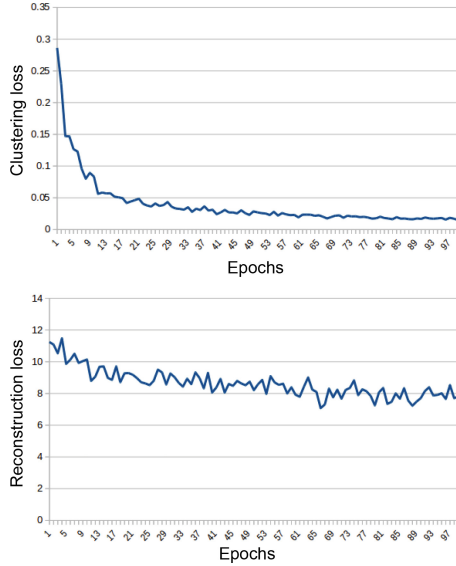


Fig. 4. Loss values over different epochs in the finetuning stage on MNIST dataset. Observe that the clustering loss decreases more compared to the reconstruction loss. This is because the reconstruction is optimized mostly in the pretraining stage. During the finetuning stage, the clustering loss plays a major role, while the reconstruction loss is mainly to keep the data representability unaffected in pursuing the goal of clustering.

We also present some graph visualizations to showcase the distribution of the points across various clusters over different epochs. From Fig. 3, we see that after pre-training, the membership values are in the low confidence region. However, as the learning progresses, the memberships are mostly in the high confidence region, indicating the learning of the clustering parameters. In Fig. 4, we present the plot of the clustering and reconstruction losses of our method over the 100

fine-tuning epochs on the MNIST dataset. Observe that the losses decrease over time, showing progress towards convergence.

4.7 Ablation Study

In this section, we conduct an ablation study to understand the significance of the centroid reinitialization (say, rein) strategy in the proposed approach. While our method shows improvement over the compared methods on the presented datasets, to understand the effect of the centroid-rein strategy, we show the results of an instance of our approach without the rein-strategy (denoted as Ours^{-rein}). The results in terms of both the metrics ACC and NMI are presented in Table 5 and 6. It can be observed that our method without the rein strategy does not improve over the compared methods. Which implies that the centroid rein strategy is crucial to the success of the proposed approach. However, if we join the centroid-rein strategy with the approach of DKM (denoted as DKM^{+rein}), no improvement is observed, which can be verified from the results presented in Table 5 and 6, where the results of DKM^{+rein} are in fact inferior to those of DKM. This shows that the usefulness of the rein strategy depends on the choice of the clustering specific loss.

Please note that, in DKM^{+rein}, we reinitialize the cluster centroids after every finetuning epoch of the clustering phase of DKM. We edited the published code of DKM¹ to get this result.

Table 5. Ablation study: understanding the role of centroid reinitialization strategy in terms of the ACC metric. Method^{+rein} and Method^{-rein} denote the method with and without reinitialization strategy, respectively.

Dataset	MNIST	USPS	COIL100	CMU-PIE	RCV1
DCN	81.10 \pm 1.90	73.0 \pm 0.80	49.23 \pm 0.88	24.78 \pm 1.66	56.70 \pm 3.60
DKM	84.00 \pm 2.20	75.7 \pm 1.30	49.50 \pm 0.78	31.61 \pm 0.86	58.3 \pm 3.80
DKM ^{+rein}	58.30 \pm 4.22	52.66 \pm 2.59	28.27 \pm 0.94	23.70 \pm 5.89	45.52 \pm 2.99
Ours ^{-rein}	85.49 \pm 4.92	70.12 \pm 1.93	42.71 \pm 6.36	15.39 \pm 1.74	58.55 \pm 0.23
Ours	90.01 \pm 5.83	78.74 \pm 4.21	51.56 \pm 0.94	32.40 \pm 1.81	60.39 \pm 2.33

Table 6. Ablation study: understanding the role of centroid reinitialization strategy in terms of the NMI metric. Method^{+rein} and Method^{-rein} denote the method with and without re-initialization strategy, respectively.

Dataset	MNIST	USPS	COIL100	CMU-PIE	RCV1
DCN	75.7 \pm 1.1	71.9 \pm 1.2	76.76 \pm 0.47	52.33 \pm 1.88	31.6 \pm 4.3
DKM	79.6 \pm 0.9	77.6 \pm 1.1	77.82 \pm 0.25	61.92 \pm 0.82	33.1 \pm 4.9
DKM ^{+rein}	49.43 \pm 3.35	46.41 \pm 2.20	55.57 \pm 0.77	44.14 \pm 9.61	16.60 \pm 2.35
Ours ^{-rein}	86.88 \pm 1.36	76.64 \pm 0.82	72.36 \pm 3.06	46.51 \pm 2.17	33.18 \pm 0.25
Ours	87.64 \pm 1.55	80.64 \pm 1.20	78.17 \pm 0.40	62.18 \pm 0.89	35.85 \pm 3.03

¹ <https://github.com/MaziarMF/deep-K-means>.

5 Conclusions and Future Works

This paper introduces a centroid-based clustering method that improves on the existing deep neural network-based K-means approaches. Along with our proposed clustering-specific loss function, we proposed the idea of employing centroid reinitialization after every finetuning epoch in the clustering phase. We have empirically shown the importance of this reinitialization. While the idea presented in this paper is verified empirically, a thorough theoretical justification is required in the future. An in-depth comparative analysis of the difference between our formulation and that of our closest deep clustering variant is worth doing in the future to gain better clarity. In the future, we also plan to extend this idea towards improving the performance of the without pre-training case. Instead of initializing the cluster centers with K-means, how a random centroid initialization can achieve similar performance with pre-training can also be a good line of research in the future.

References

1. De Soete, G., Carroll, J.D.: K-means clustering in a low-dimensional Euclidean space. In: Diday, E., Lechevallier, Y., Schader, M., Bertrand, P., Burtschy, B. (eds.) *New Approaches in Classification and Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 212–219. Springer, Heidelberg (1994). https://doi.org/10.1007/978-3-642-51175-2_24
2. Fard, M.M., Thonet, T., Gaussier, E.: Deep k-means: jointly clustering with k-means and learning representations. *Pattern Recogn. Lett.* **138**, 185–192 (2020)
3. Feng, Q., Chen, L., Chen, C.L.P., Guo, L.: Deep fuzzy clustering—a representation learning approach. *IEEE Trans. Fuzzy Syst.* **28**(7), 1420–1433 (2020)
4. Genevay, A., Dulac-Arnold, G., Vert, J.-P.: Differentiable deep clustering with cluster size constraints. *arXiv preprint arXiv:1910.09036* (2019)
5. Guo, X., Gao, L., Liu, X., Yin, J.: Improved deep embedded clustering with local structure preservation. In: *IJCAI*, pp. 1753–1759 (2017)
6. Jiang, Y., Xu, Q., Yang, Z., Cao, X., Huang, Q.: DM2C: deep mixed-modal clustering. *Adv. Neural Inf. Process. Syst.* **32** (2019)
7. Kuhn, H.W.: The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**(1–2), 83–97 (1955)
8. Ma, Z., Kang, Z., Luo, G., Tian, L., Chen, W.: Towards clustering-friendly representations: subspace clustering via graph filtering. In: *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 3081–3089 (2020)
9. Park, S., et al.: Improving unsupervised image clustering with robust learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12278–12287 (2021)
10. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008)
11. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: *International Conference on Machine Learning*, pp. 478–487. PMLR (2016)

12. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards K-means-friendly spaces: simultaneous deep learning and clustering. In: International Conference on Machine Learning, pp. 3861–3870. PMLR (2017)
13. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5147–5156 (2016)



PulmoNetX: A Hybrid Vision Transformer Approach for Multi-scale Spatial Feature Reduction in Pneumonia Classification

Asifuzzaman Lasker¹(✉), Mridul Ghosh², Sk Md Obaidullah¹,
Chandan Chakraborty³, Kaushik Roy⁴, and Umapada Pal⁵

¹ Department of Computer Science and Engineering, Aliah University,
Kolkata 700160, India
asifuzzaman.lasker@gmail.com

² Department of Computer Science, Shyampur Siddheswari Mahavidyalaya,
Howrah 711312, India

³ Department of Computer Science and Engineering, National Institute of Technical
Teachers' Training and Research, Kolkata 700106, India
chandan@nitttrkol.ac.in

⁴ Department of Computer Science, West Bengal State University, Barasat 700126,
India

⁵ Computer Vision and Pattern Recognition Unit, Indian Statistical Institute,
Kolkata, India
umapada@isical.ac.in

Abstract. An innovative deep learning structure, PulmoNetX, integrates the capabilities of Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) to enhance pneumonia detection in chest X-ray imagery. During preprocessing, images are normalized in size, converted to grayscale, and subjected to contrast amplification to emphasize essential features. PulmoNetX employs a hybrid methodology to capture both the local and global characteristics of images, leading to significant advancements in diagnosing different pneumonia types, such as COVID-19-induced, viral, and bacterial pneumonia. Comparative studies reveal that PulmoNetX surpasses leading Vision Transformer models in terms of precision, recall, F1-score, and overall accuracy, highlighting its advanced processing abilities and its promise as an effective diagnostic tool in X-ray lung disease detection.

Keywords: Deep learning · Pneumonia · Chest X-ray · CNN · Vision Transformer

1 Introduction

Pneumonia represents a significant global health burden, with diverse etiologies including bacterial, viral, and fungal infections [1]. The outbreak of COVID-19

has further complicated the landscape, highlighting the need for accurate differentiation between pneumonia types for effective treatment and management. The complexity of pneumonia, coupled with its overlapping clinical and radiographic characteristics, necessitates advanced diagnostic tools for precise classification and intervention.

The emergence of computer vision coupled with artificial neural network (deep learning) has revolutionized medical imaging, enabling the extraction of detailed features and patterns from complex datasets [2]. In the realm of pulmonary diseases, these technologies have facilitated the development of automated systems capable of analyzing chest radiographs and CT scans with high accuracy. Deep learning models, particularly CNNs, have been instrumental in identifying and classifying various forms of pneumonia, thus aiding in the rapid and reliable diagnosis of this condition.

Vision Transformers (ViTs) have risen as a compelling counterpart to Convolutional Neural Networks (CNNs), presenting unique benefits in the realm of medical image analysis [3]. ViTs excel in capturing long-range dependencies and global context within images, which are often missed by the local receptive fields of CNNs. This capability enables ViTs to better understand the holistic nature of medical imagery, leading to more accurate and comprehensive analyses of complex diseases like pneumonia.

Combining a comprehensive view of pneumonia with the strengths of computer vision, deep learning techniques, and the innovative approaches of Vision Transformers (ViTs) provides a deeper insight into the classification and semantic segmentation of lung [4]. The synergy between these domains facilitates the development of more sophisticated and accurate diagnostic models. This integration highlights the importance of leveraging both global and local features in medical imaging to enhance diagnostic accuracy and efficiency in detecting and classifying pneumonia.

Contributions of PulmoNetX:

Advanced Hybrid Architecture: PulmoNetX integrates depthwise and pointwise convolutions with Vision Transformers, creating a hybrid model that captures both local and global features of pulmonary images. This architecture facilitates precise classification of pneumonia types by leveraging the strengths of CNNs for detailed feature extraction and ViTs for comprehensive spatial analysis.

Enhanced Diagnostic Accuracy: Through its innovative use of channel-wise filters and depthwise convolutions, PulmoNetX enhances image quality and feature representation, leading to improved detection and differentiation of pneumonia conditions, including Normal, COVID-19, Viral Pneumonia, and Bacterial Pneumonia.

Efficient Computational Performance: By utilizing depthwise convolution layers, the architecture enhances computational efficiency. This approach decreases both the parameter count and the computational burden while maintaining the capability to analyze and interpret intricate image features, rendering it ideal for real-time analysis in medical imaging.

2 Methodology

In our study, centered on the PulmoNetX model, delineates the comprehensive approach undertaken to enhance pneumonia classification accuracy through advanced deep learning techniques. At the core of this research is the development of a hybrid architecture that synergistically combines the spatial feature extraction capabilities of Convolutional Neural Networks (CNNs) with the dynamic scaling and global context assimilation facilitated by Vision Transformers (ViTs). This methodological framework is meticulously designed to tackle the inherent complexities of pulmonary imaging, addressing the multifaceted nature of pneumonia, which spans various types and etiologies, including COVID-19, Viral Pneumonia, and Bacterial Pneumonia. The ensuing sections will explicate the technical processes, from image preprocessing and data augmentation to the intricacies of the model architecture, ultimately culminating in the deployment of PulmoNetX for clinical evaluation and validation.

2.1 Image Pre-processing

During the preliminary phase of preprocessing, each X-ray image was resized to maintain a consistent dimension of 256 by 256 pixels to accommodate the variation in sizes present in the dataset. For further filtering process all images transform to grayscale. conversion highlights the intensity variations within the image, which are more relevant for detecting anomalies in lung tissues than color information. By representing the image in shades of gray, it becomes easier to assess and interpret the various densities and structures within the lungs, such as the bronchial walls, airspaces, and lung parenchyma.

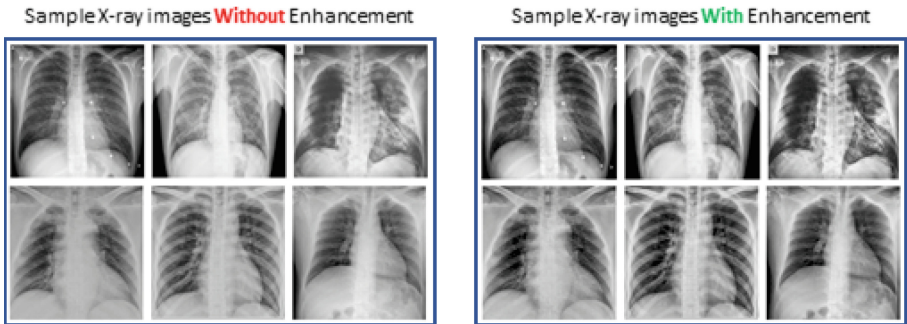


Fig. 1. Generate a pair of chest X-ray images side by side for different medical conditions. On the left, an unenhanced X-ray image, and on the right, the same X-ray but enhanced to highlight details. The conditions should vary, showcasing different classes like pneumonia, a collapsed lung, and a healthy lung for comparison.

Following grayscale conversion, histogram equalization is performed to enhance the contrast of the grayscale image. This process redistributes the intensity values of the image, allowing for a more uniform and enhanced visibility of

lung structures and potential pathological findings. By adjusting the contrast, histogram equalization makes subtle differences in tissue density more apparent, which is crucial for detecting lesions, nodules, or other abnormalities. Subsequently, edge detection is carried out to identify the boundaries within the lung tissues. Using techniques like the Sobel operator, edge detection emphasizes the edges and contours of structures within the X-ray, aiding in the delineation of lung anatomy and potential pathological changes. This step is essential for enhancing the visibility of the lung’s structural boundaries which depicted in the Fig. 1, thereby facilitating more accurate diagnosis and analysis in the subsequent stages of image classification.

The pairplot in Fig. 2 illustrates the improvement in image quality after pre-processing across four different classes. Specifically, it presents SSIM and PSNR values for four disease categories—COVID-19, Normal, Bacterial Pneumonia, and Viral Pneumonia—offering a detailed visualization of the enhancement in image quality. The diagonal plots, which display the distribution of SSIM and PSNR values for each disease category, reveal that Normal images have the highest and most consistent SSIM values, peaking around 0.98, while COVID-19 images have a broader and lower peak around 0.90, indicating lower structural similarity. Similarly, Normal images exhibit the highest PSNR values, indicating superior noise reduction, whereas COVID-19 images show the lowest and most variable PSNR values, reflecting challenges in noise reduction.

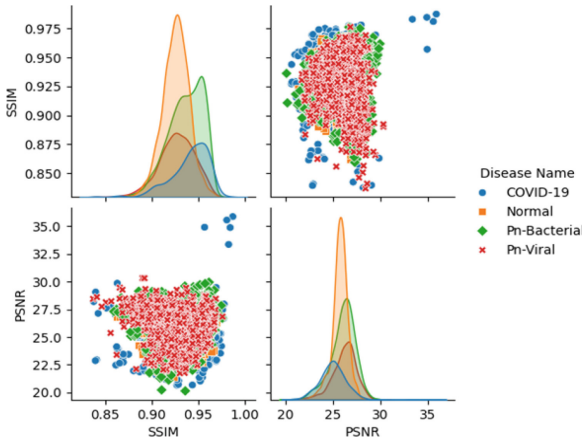


Fig. 2. Pairplot of SSIM and PSNR values by Diseases.

The off-diagonal scatter plots illustrate the relationship between SSIM and PSNR for each disease category. These plots show that Normal and Pneumonia-Bacterial images cluster in the high SSIM and PSNR range, suggesting that these images maintain both high structural similarity and effective noise reduction. In contrast, Pneumonia-Viral images show moderate levels of these metrics with greater variability, and COVID-19 images display the widest spread and

lowest values for both SSIM and PSNR. This analysis highlights the need for tailored enhancement strategies to improve image quality, particularly for COVID-19 and Pneumonia-Viral images, which currently exhibit lower performance in structural similarity and noise reduction.

2.2 Architecture Preparation

The proposed model architecture is designed to effectively classify lung diseases from preprocessed X-ray images. It integrates depthwise convolutional layers, custom combined layers, and a modified Vision Transformer (ViT) to capture complex patterns and features of the lung X-ray images.

In the Fig. 3 depicted the architecture in two main modules: the Combine Layer (CL) module and the Transformer module. The primary structure begins with an input layer, followed by a series of CL layers (highlighted in blue). Initially, the input layer feeds into the first CL layer, which processes the input dimensions. The output features from the first CL layer are then fed into the second CL layer, continuing in a similar fashion through subsequent CL layers. After extracting relevant features from these CL layers, the output of the last CL layer is patchified and passed to the Patch Encoder block to identify specific patches. This patchified step is useful because it reduces computational complexity by converting the 2D spatial data of an image into a manageable sequence of patch embeddings, making it feasible to process with Transformers. Additionally, it enables the model to capture long-range dependencies and relationships across the image, which enhances its ability to understand global context and intricate patterns. The modified Transformer module (highlighted in yellow) is then utilized to generate global features. These encoded patches are fed into a Transformer module, which includes multi-head attention (mha) and multi-layer perceptron (mlp) components to capture intricate spatial dependencies and global context. Finally, the processed features are flattened and passed through a dense network for classification, leading to the prediction of the diagnostic outcome. The architecture effectively integrates convolutional neural network (CNN) elements with Transformer modules to leverage both local and global feature representations for improved diagnostic accuracy. In the following paragraphs, we discuss the architecture in more detail using mathematical notation.

1. **Depthwise Convolutional Layers:** Depthwise convolutional layers are utilized to perform spatial filtering while reducing the number of parameters and computational cost. For an input feature map X , a depthwise convolution operation with a kernel K is applied as follows:

$$X_{out} = X_{depth} * K$$

where $*$ denotes the convolution operation. This is followed by batch normalization and GELU (Gaussian Error Linear Unit) activation to stabilize the learning and introduce non-linearity.

2. **Custom Combined Layers:** The model employs a series of combined layers where each layer consists of a depthwise convolutional block followed by a pointwise convolutional block to enhance feature extraction. For a given input X , the combined layer operation can be represented as:

$$X_{combo} = \text{Conv}(\text{DepthwiseConv}(X))$$

where Conv and DepthwiseConv denote the pointwise and depthwise convolution operations, respectively.

3. **Patch Encoding and Vision Transformer:** The processed feature maps are then divided into patches and encoded for the transformer network. If P denotes the patch extraction operation on the feature map X_{combo} , then the patch encoding can be mathematically represented as:

$$E = \text{Encode}(P(X_{combo}))$$

where E represents the encoded patches ready for transformer processing. The Vision Transformer then processes these encoded patches through a series of self-attention (Att) and multi-layer perceptron (MLP) layers to capture global dependencies and relationships between patches.

4. **Transformer Mechanism:** The transformer architecture utilizes self-attention mechanisms to process the encoded patches. For a set of encoded patches E , the transformer applies a series of attention operations Att and MLP blocks MLP , followed by layer normalization (LN), as given by:

$$E' = \text{LN}(\text{MLP}(\text{LN}(\text{Att}(E) + E)) + \text{Att}(E))$$

This process enables the model to effectively integrate information from different parts of the image, leading to a comprehensive understanding of the lung X-ray features.

5. **Classification Head:** The output of the transformer is passed through a classification head, typically consisting of a flatten operation, dropout layers, and dense layers to predict the lung disease categories. The final output Y representing the disease classification is obtained by:

$$Y = \text{Dense}(\text{Dropout}(\text{Flatten}(E')))$$

where the Dense layer is configured to match the number of target classes in the lung disease classification task.

This model architecture, combining depthwise convolutional layers, custom combined layers, and a modified Vision Transformer, is tailored to capture both local and global features in lung X-ray images, facilitating accurate and robust classification of lung diseases.

3 Experiment

This section delineates the methodological framework of our study, including the experimental setup, the data utilized, the metrics adopted for evaluating

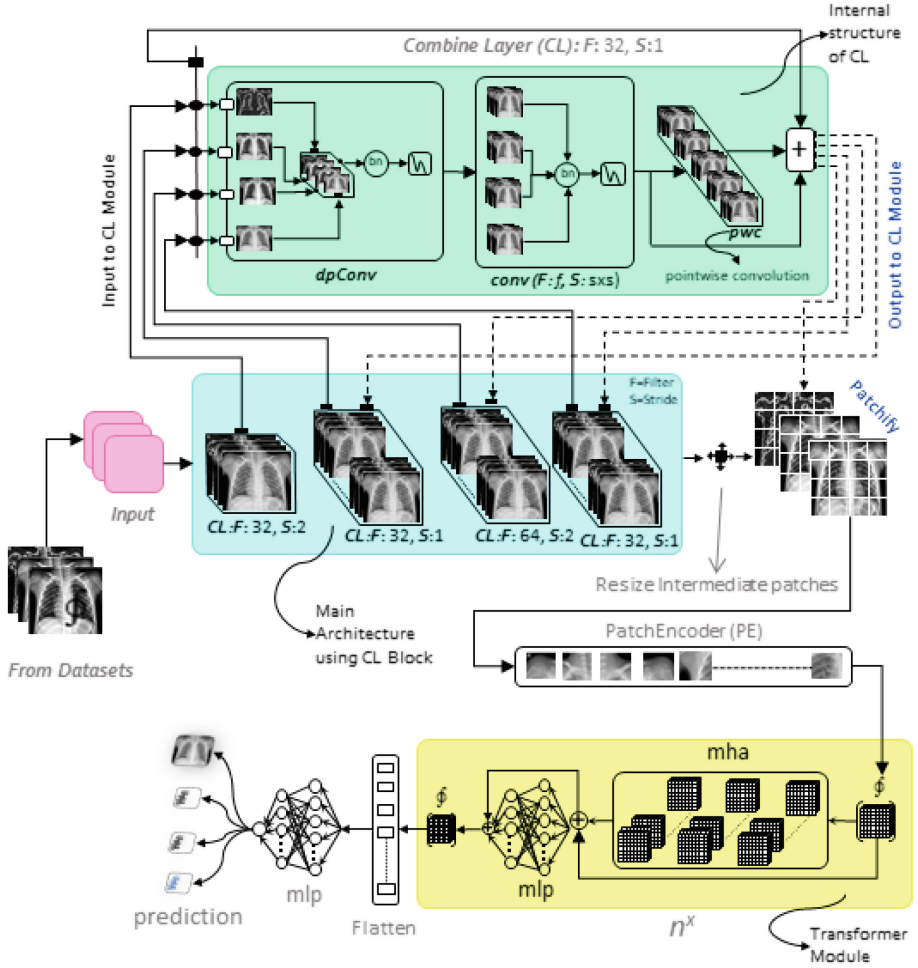


Fig. 3. PulmoNetX: The Architecture for Pneumonia Classification (Color figure online)

performance, and the empirical outcomes. It provides a detailed account of how the experiments were conducted, the nature of the dataset examined, the criteria used to assess the model's efficacy, and the results derived from these evaluations.

3.1 Datasets

The study used a dataset combined from 15 sources of PA Chest X-rays, totaling 9,208 images, categorized into Normal, Viral Pneumonia, Bacterial Pneumonia, and COVID-19 classes. Originally, the dataset had 20,226 samples [5], but was curated to remove redundancies using cosine similarity via the InceptionV3 architecture, resulting in 1,281, 3,001, 1,656, and 3,270 samples for each

category, respectively. To prevent data leakage a common issue leading to overly optimistic training results but poor testing performance the dataset was carefully prepared and divided into separate training and testing sets, maintaining proper class distributions.

3.2 Experimental Setup

For the trial of the PulmoNetX model, we utilized a computer with specifications that fall within the mid-range category. The configuration comprised a system with 32 gigabytes of memory, powered by an Intel Core i5 CPU, complemented by a dedicated graphics processing unit to boost the efficiency of neural network training. The computational performance was further enhanced by the use of cuDNN. We operated the experiments on a stable platform provided by Windows 10 Pro, specifically version 20H2. For the development and evaluation of our methods, TensorFlow version 2.0.0 was employed as the framework for deep learning.

3.3 Evaluation Protocols

In the healthcare sector, the evaluation of machine learning models requires robust validation measures due to the critical nature of medical decision-making. Alongside accuracy, we have expanded our evaluation framework beyond mere accuracy to include metrics like precision, recall, F1 score, as well as the area under the curve (AUC) and receiver operating characteristic (ROC) curves because relying solely on accuracy is not sufficient [4,6]. These metrics are computed from the elemental components of binary classification outcomes, namely True Ω and False Ξ Positives, also False Ψ Negatives and True Θ Negatives. Accuracy is determined by the following formula 1:

$$Accuracy(A) = \frac{\Omega + \Theta}{\Omega + \Theta + \Xi + \Psi} \quad (1)$$

Accuracy reflects the proportion of accurate predictions relative to the overall predictions conducted. To complement accuracy, we consider precision, recall, and the F1 score, detailed in Eqs. 2, 3, and 4 respectively.

$$Precision(P) = \frac{\Omega}{\Omega + \Xi} \quad (2)$$

$$Recall(R) = \frac{\Omega}{\Omega + \Psi} \quad (3)$$

In the realm of healthcare, a high number of False Positives Ξ can impose undue strain on medical infrastructure by erroneously flagging healthy individuals as diseased. Conversely, the occurrence of False Ψ Negatives holds significant implications in medical diagnostics; thus, recall is a crucial metric. A lower recall indicates potential risks of misdiagnosis and inadequate patient treatment.

$$F1 - Score = 2 \left(\frac{\Pi \cdot P}{\Pi + P} \right) \quad (4)$$

Focusing disproportionately on False Positives Ξ or False Negatives Ψ may cause an oversight of the alternate error type. Therefore, we utilize the F1 score as a metric that establishes equilibrium between Precision Π and Recall (P), thus ensuring a comprehensive assessment that accounts for both types of errors.

3.4 Ablation Study

In the ablation study, we adjusted four key parameters: Image size, Patch size, Projection dimension, and Batch size. For the image size, dimensions of 64 and 128 were tested. Increasing the resolution improved accuracy, but at a resolution of 512, the accuracy only slightly improved and led to significant overfitting; therefore, we selected an image size of 256. The patch size, an intermediate layer of the PulmoNetX model, was tested at 4, 8, 16, 32, and 64, with the 16-value patch size yielding the highest accuracy for an image size of 256. Regarding the projection dimension, 128 provided the best accuracy compared to other tested values. We fixed the batch size at 16, as larger batch sizes did not enhance accuracy and increased computational intensity. The final settings of our model are presented in Table 1.

Table 1. Extended Vision Transformer Model Training Parameters

Parameters	Values
Image size	256
Patch size	32
Projection dimension	128
Batch size	16
Epochs	50
Train Test Split	80% - 20%
Optimizer	adam
Learning Decay	cosine restart

Table 2 presents a comparison of evaluation metrics for a model tested with and without image enhancement. The standard metrics of classification is a noticeable improvement when image enhancement techniques are applied. The most significant increase is observed in Accuracy, which goes from 0.8772 to 0.8924 with enhancement.

Table 2. Evaluation metrics results of without image enhancement and with image enhancement.

Metrics	Without Enhancement	With Enhancement
Precision	0.8821	0.8927
Recall	0.8804	0.8924
F1-score	0.8782	0.8882
Accuracy	0.8772	0.8924

4 Results and Discussions

In this section, we delineate the performance metrics of the PulmoNetX architecture, contrasting its classification capabilities against a dataset consisting of Normal, Viral Pneumonia, Bacterial Pneumonia, and COVID-19 cases. The evaluation is articulated through the examination of loss and accuracy progressions, ROC curve analysis, and confusion matrix insights. Additionally, a comparative analysis positions PulmoNetX alongside leading Vision Transformer (ViT) models, showcasing its efficacy through both quantitative metrics and statistical analysis.

During the initial training of PulmoNetX, as shown in Fig. 4, the loss rapidly decreased in early epochs, highlighting efficient learning. This decline stabilized after about 15 epochs, indicating optimal parameter tuning. Concurrently, validation loss followed a similar trend, with minor fluctuations suggesting good generalization to new data.

In the accuracy assessment phase, represented in Fig. 4 (right), training accuracy quickly reached a stable high level, demonstrating effective pattern retention and application. Validation accuracy closely matched this, showing only minor deviations, which underscores the PulmoNetX model’s consistency and reliability across different data sets, a key trait for clinical diagnostics..

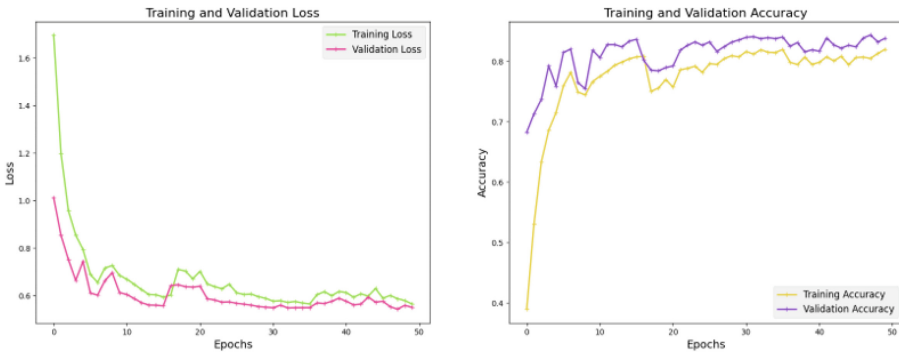


Fig. 4. Training and validation loss (left) and accuracy (right) over 50 epochs.

The confusion matrix, represented in Fig. 5, further elucidates the classification prowess of PulmoNetX, showcasing a high number of true positives and a limited number of misclassifications. The architecture demonstrates exceptional precision in distinguishing between COVID-19 and normal cases, with only minimal confusion with other pneumonia types.

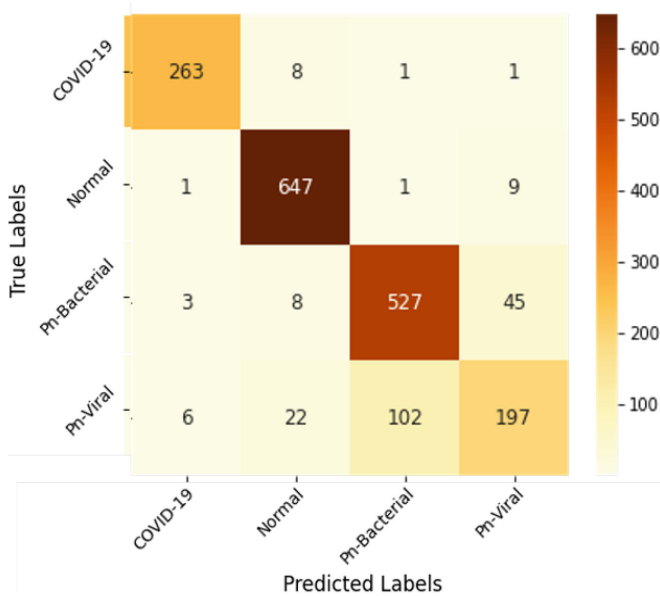


Fig. 5. The matrix displays PulmoNetX’s classifications for X-ray images into Normal, Viral Pneumonia, Bacterial Pneumonia, and COVID-19, detailing correct predictions, incorrect predictions, correct rejections, and incorrect rejections for each category.

Figure 6 presents the ROC curves for the multi-class classification problem. The area under the ROC curve (AUC) for classes indicative of Normal (Class 0), Viral Pneumonia (Class 1), Bacterial Pneumonia (Class 2), and COVID-19 (Class 3) show high diagnostic ability, with AUC values of 0.98, 0.98, 0.91, and 0.78, respectively. These values confirm the model’s robustness in distinguishing between the various classes, with particularly high performance in differentiating Normal and Viral Pneumonia cases

The performance of the PulmoNetX architecture was benchmarked against a variety of state-of-the-art Vision Transformer (ViT) models to validate its efficacy in the classification of pulmonary conditions. As indicated in Table 3, the comparison encompassed several metrics critical to classification tasks: precision, recall, F1-score, and overall accuracy.

PulmoNetX distinguishes itself by delivering a precision of 0.8927, indicating that when it predicts any class, it is correct approximately 89.27% of the time. Its recall of 0.8924 implies that it identifies 89.24% of all relevant instances

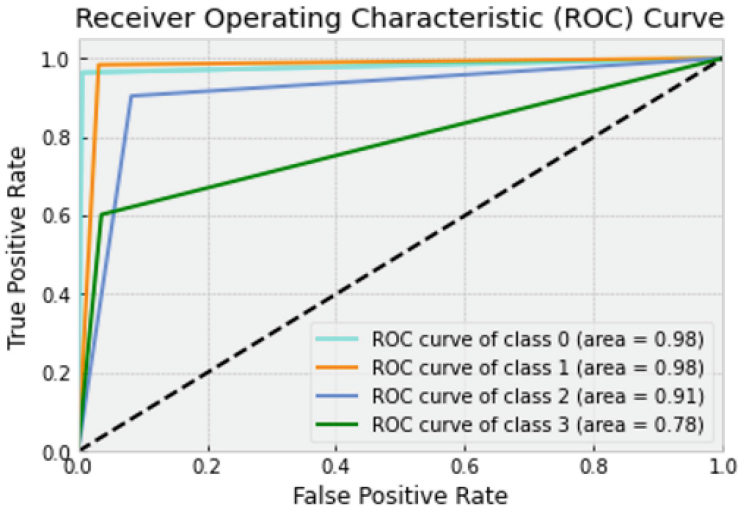


Fig. 6. ROC curves for the multi-class classification.

within the dataset correctly. The F1-score, a harmonic mean of precision and recall, stands at 0.8882, suggesting a balanced classification performance, particularly important in medical diagnostic tasks where both false positives and false negatives carry significant consequences.

The overall accuracy of PulmoNetX, calculated at 89.24%, surpasses its closest competitor, the DeepViT model, by a margin of 2.49 percentage points. This margin may appear modest; however, in the domain of medical image classification, even slight improvements can translate into substantial clinical benefits.

A statistical analysis was conducted to assess the significance of the differences in performance metrics between PulmoNetX and the other models. Using a paired t-test, the differences in accuracy rates were found to be statistically significant with p-values below the 0.05 threshold when compared against all models except for DeepViT, for which the p-value was marginally above the threshold, indicating a suggestive but not conclusive difference in performance.

The superior performance of PulmoNetX can be attributed to its enhanced feature extraction capabilities and optimized attention mechanisms, which allow for a more nuanced understanding of complex pulmonary patterns in imaging data. Notably, the architecture's ability to maintain a high F1-score across multiple classes suggests that it does not sacrifice precision for recall or vice versa, which is often observed in models with unbalanced class distributions.

Furthermore, the performance advantage held by PulmoNetX may also be partially due to its robust training regime, which included advanced data augmentation techniques and a carefully curated dataset that represented a wide spectrum of pathological presentations, as visualized in the confusion matrix (Fig. 5). The high diagonal values in the confusion matrix corroborate the

Table 3. Performance Evaluation of PulmoNetX Architecture against State-of-the-Art Vision Transformer (ViT) Models.

Models name	Precision	Recall	F1-score	Accuracy
ViT [7]	0.8518	0.8533	0.8515	85.33
DeepViT [8]	0.8566	0.8474	0.8512	86.75
LeViT [9]	0.1277	0.3574	0.1882	35.74
CCT [10]	0.8077	0.7968	0.7745	79.68
RegionViT [11]	0.8210	0.8115	0.8133	81.15
MobileViT [12]	0.7486	0.7463	0.7372	77.85
Cross ViT [13]	0.4854	0.5845	0.5212	58.45
PulmoNetX (ours)	0.8927	0.8924	0.8882	0.8924

model’s precision and recall values, indicating a strong true positive rate across the board.

The statistical and empirical evidence collectively point to the PulmoNetX architecture as a significant advancement in the field of pulmonary disease classification using deep learning. Its performance metrics, backed by statistical validation, underscore its potential as a reliable tool for aiding medical professionals in diagnosing various conditions, including the challenging differentiation of COVID-19 from other types of pneumonia.

4.1 Potential Limitations

The model’s computational complexity requires significant resources for training and inference, which may limit its deployment in resource-constrained environments. Additionally, the model’s performance is highly dependent on the quality and diversity of the training data, potentially affecting its generalizability to different populations and imaging conditions. The sensitivity of preprocessing techniques, such as histogram equalization and edge detection, to parameter settings also poses a challenge. Future work will focus on optimizing these parameters, enhancing data augmentation strategies, simplifying the model architecture to reduce computational demands, and validating the model in diverse clinical settings to improve its robustness and applicability.

5 Conclusion

PulmoNetX represents a significant advancement in the automated classification of pulmonary diseases using deep learning technologies. By effectively merging depthwise and pointwise convolutions with Vision Transformers, PulmoNetX captures nuanced spatial features of pulmonary images, leading to precise

and reliable classification outcomes. The model not only excels in differentiating between Normal, COVID-19, Viral Pneumonia, and Bacterial Pneumonia but also showcases computational efficiency and robustness against overfitting. Statistical analyses confirm the model's superior performance compared to established Vision Transformer models. Future work will focus on extending PulmoNetX's applicability to other medical imaging modalities and exploring its potential in real-world clinical settings to enhance diagnostic accuracy and patient care.

References

1. World Health Organization: Pneumonia (n.d.). <https://www.who.int/news-room/fact-sheets/detail/pneumonia>
2. Lasker, A., Ghosh, M., Obaidullah, S.M., Chakraborty, C., Roy, K.: LWSNet-a novel deep-learning architecture to segregate Covid-19 and pneumonia from X-ray imagery. *Multimedia Tools Appl.* **82**(14), 21801–21823 (2023)
3. Shamshad, F., et al.: Transformers in medical imaging: a survey. *arXiv* 2022. *arXiv preprint* [arXiv:2201.09873](https://arxiv.org/abs/2201.09873)
4. Lasker, A., Ghosh, M., Obaidullah, S.M., Chakraborty, C., Goncalves, T., Roy, K.: Ensemble stack architecture for lungs segmentation from X-ray images. In: Yin, H., Camacho, D., Tino, P. (eds.) *IDEAL 2022. LNCS*, vol. 13756, pp. 3–11. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-21753-1_1
5. Sait, U., et al.: Curated dataset for COVID-19 posterior-anterior chest radiography images (X-Rays). *Mendeley Data*, 1 (2020)
6. Ghosh, M., Roy, S.S., Mukherjee, H., Obaidullah, S.M., Gao, X.Z., Roy, K.: Movie title extraction and script separation using shallow convolution neural network. *IEEE Access* **9**, 125184–125201 (2021)
7. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint* [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
8. Zhou, D., et al.: DeepViT: towards deeper vision transformer. *arXiv preprint* [arXiv:2103.11886](https://arxiv.org/abs/2103.11886) (2021)
9. Graham, B., et al.: LeViT: a vision transformer in convnet's clothing for faster inference. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12259–12269 (2021)
10. Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J., Shi, H.: Escaping the big data paradigm with compact transformers. *arXiv* 2021. *arXiv preprint* [arXiv:2104.05704](https://arxiv.org/abs/2104.05704)
11. Chen, R., Panda, R., Fan, Q.: U.S. Patent No. 11,915,474. Washington, DC: U.S. Patent and Trademark Office (2024)
12. Mehta, S., Rastegari, M.: MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint* [arXiv:2110.02178](https://arxiv.org/abs/2110.02178) (2021)
13. Chen, C. F. R., Fan, Q., Panda, R.: CrossViT: cross-attention multi-scale vision transformer for image classification. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 357–366 (2021)



Federated K-Means Clustering

Swier Garst^(✉)  and Marcel Reinders 

Delft University of Technology, Delft, The Netherlands
{S.J.F.Garst,M.J.T.Reinders}@tudelft.nl

Abstract. Federated learning is a technique that enables the use of distributed datasets for machine learning purposes without requiring data to be pooled, thereby better preserving privacy and ownership of the data. While supervised FL research has grown substantially over the last years, unsupervised FL methods remain scarce. This work introduces an algorithm which implements K-means clustering in a federated manner, addressing the challenges of varying number of clusters between centers, as well as convergence on less separable datasets.

Keywords: Federated Learning · K-Means clustering · Distributed machine learning

1 Introduction

Nowadays, lots of data is being generated in a distributed fashion. Mobile phones and other personal devices such as smart watches enable the collection of massive amounts of data. If made accessible, this data could prove useful for improving the performance of the services provided by these devices. However, due to a growing concern on data privacy, more and more users of these devices are hesitant in sharing their data. Furthermore, regulations such as the General Data Protection and Regulation (GDPR) act prevent the collection of data of this kind in bulk. Federated learning (FL) ([15]) was introduced as a solution to this problem. In short, instead of pooling data to train a single model, instances of a model are being shared to data owners (clients), which then train the model on their local data. Then, these trained models are sent back to the central server, which aggregates the results. Next, a new round begins with the server sending out the updated models. This cycle continues until convergence.

Over the past couple of years, research has shown FL to be a promising technique, reaching performances comparable to a central approach in which all data of the clients is pooled at a single location [11, 17]. The vast majority of the federated learning research has been focusing on the supervised learning paradigm. Little work has been done on unsupervised federated learning methods, even less so when specifically looking into clustering techniques [12, 13]. One of these

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78166-7_8.

clustering techniques is k-means clustering [7]. In a federated learning setting, k-means clustering can be described as trying to find overarching cluster means over data which is distributed among different datasets (clients).

Prior work has been done on creating federated k-means clustering algorithms [5, 8, 10, 14, 18]. We focus on solving the issue of a variable amount of local clusters. When data distributions between clients differ, it is likely that not all clients share data from all global clusters. When this is the case, the amount of clusters present per client can differ. This complicates federated clustering in two ways. First, matching cluster means between clients becomes less straightforward, as there is no one-to-one matching anymore. Second, when the local k does not correspond to the global k , the problem of finding an optimal k scales linearly with the amount of clients. Furthermore, when clients hold only part of the data, it can be hard to distinguish between outliers and samples of a different cluster, without the knowledge of the data distribution of other clients. Altogether, this makes manually determining a value for k on each client locally complicated, if not infeasible without loss of performance.

We propose an iterative federated k-means clustering algorithm (FKM) that automatically determines the local value for k . By iteratively aggregating local cluster means, and running k-means locally on all clients in parallel, we are able to create a clustering that in many cases corresponds to a central clustering, i.e. the k-means clustering that would occur if all data was pooled together. By pruning empty global clusters on local clients, we are able to deal with a variable amount of local clusters between clients, without having to set the values for k locally on each client.

2 Related Work

Since its inception in 2017, federated learning has been applied in various scenarios. The most well known taxonomy of federated learning systems is the split between the cross-device and cross-silo settings [12]. In brief, in the cross-device setting, many, in the order of thousands or more, devices are connected. However their connection is usually unstable, and cross-device federations will have to deal with users dropping out or joining later throughout the process. This setting is mostly applicable to federations with end-user devices, such as mobile phones. On the other hand, in the cross-silo setting, there are usually only a couple to tens of clients. These clients are usually quite stable and can be assumed to be connected from the beginning of the process all the way to the end. This setting is more applicable to health care centers or companies learning a shared model without sharing their data due to privacy or competition reasons.

Different challenges emerge based on whether the federation is cross-device or cross-silo and as such different algorithms have emerged for either setting. For the cross-device setting, Liu et al. introduced a method for federated k-means in a setting where each client is seen as a single sample [14]. More specifically, they tackle the issue of proactive caching in cellular networks, i.e. trying to predict what data to keep local based on popularity. Their method is robust

against user change, and assumes a tiered architecture with base stations and sub stations, as can be found in next-generation cellular networks. Kumar et al. also propose a federated k-means clustering algorithm that can be applied for the cross-device setting [10]. Their method assumes a central dataset available at the server, on which a k-means clustering is pre-trained. It is then distributed across all clients who update the clustering based on their local data, after which the clustering is aggregated again on the server. Finally, Hou et al. created a k-means clustering algorithm based on homomorphic encryption and blockchain [8]. Although their method is not explicitly applied to the cross-device setting, it does share encrypted versions of the data, which is often still unacceptable for many cross-silo use-cases.

For the cross-silo setting, Servetnyk et al. proposed a federated k-means clustering algorithm based on dual averaging and self-organising maps [18]. Although their algorithm is capable of dealing with heterogeneous data, they do not explicitly address the challenge of a variable local k , or cluster alignment. Finally, Dennis et al. propose a one-shot federated k-means clustering algorithm, which only needs one local clustering, as well as one global aggregation step [5]. As we focus mostly on the cross-silo setting, our algorithms bears the most resemblance to the method by Dennis et al. (kFed). The key differences are that our method does not require setting a local value for k on each client, which, as we argue in the introduction, can be difficult if not impossible without loss of performance. This also allows us to have a different k for each client, further improving performance. Finally, we show that, for less separable datasets, there is a substantial performance gain in iterating between the local data and global aggregation more than once, as is done by Dennis et al.

Table 1. Notations used

symbol	description
X_i	data on client i
K_g	global number of clusters
K_i	number of clusters on client i
C_g	global cluster means
C_i	cluster means of client i
M	total amount (sum) of local clusters
S_i	amount of samples for each cluster on client i
N	total amount of clients

3 Methods

Notation used throughout this section is found in Table 1. The pseudocode for our proposed federated k-means algorithm (FKM) can be found in Algorithm 1.

Algorithm 1. The federated kmeans algorithm**Input:** K_g

```

1: Init:
2: on each client  $i \in N$  do:
3:    $K_i = K_g$ 
4:    $S_i, C_i = \text{kmeans++\_init}(X_i, K_i)$   $\triangleright$  get cluster means using kmeans++ initialization
5:   send  $S_i, C_i$  to server
6: For each round  $r$  do:
7:   On server do:
8:      $C_l = [C_1|C_2|\dots|C_M]$   $\triangleright$  Concatenate all local cluster means
9:      $S_l = [S_1|S_2|\dots|S_M]$   $\triangleright$  Repeat for sample amounts per cluster
10:     $C_g = \text{kmeans}(C_l, K_g, \text{weights} = S_l)$   $\triangleright$  Obtain new global clusters using kmeans
11:    send  $C_g$  to all clients
12:   On each client  $i \in N$  do:
13:      $C_i = C_g$ 
14:      $S_i = \text{kmeans\_assign}(X_i, C_i)$   $\triangleright$  Determine empty clusters
15:      $C_i = C_i[s \neq 0 \text{ for } s \text{ in } S_i]$   $\triangleright$  Drop empty global clusters
16:      $K_i = \text{size}(C_i)$ 
17:      $S_i, C_i = \text{kmeans}(X_i, K_i, \text{init} = C_i)$   $\triangleright$  run kmeans from remaining global cluster means
18:     send  $S_i, C_i$  to server

```

The algorithm can be divided into two parts: an initialization step, in which we generate initial cluster means on each client using k-means++ initialization [1], and an iterative k-means step in which clients communicate their cluster means to the server, which aggregates these means into a ‘global’ set of means, which then gets redistributed to the clients for the next k-means iteration. See supplement A for background on k-means and k-means++.

3.1 Determining the Amount of Local Clusters

While the global amount of clusters is set (main parameter k of the k-means procedure), it is not a given that each client has data for each of these clusters. In other words, the number of clusters between clients can differ, and is not necessarily equal to the number of clusters in the pooled data. In order to solve this problem, we determine which global clusters correspond to local data in each round on each client. Before a client applies a new k-means step locally, it assigns its data to the global cluster means it has received (line 14). Next, clients check if there are empty clusters, i.e. cluster means which did not get any points assigned to them. If so, clients discard these empty clusters (line 15). The remaining (global) cluster means are then used as initialization for the next local k-means step (line 16). This way, k can locally become smaller when running k-means on the clients. Since this pruning step only happens after global aggregation, we guarantee that the discarded clusters are indeed corresponding to clusters on other clients.

3.2 Cluster Alignment

After each client has calculated one iteration of k-means (not until convergence, to avoid local minima) on their local data (each with their own amount of local clusters), they send their cluster means as well as the amount of samples per cluster back to the server. The server then concatenates all cluster means, and aggregates them. It does so by running a k-means clustering on the received local means until convergence (using the global k parameter), to align clusters from different clients to each other. This global k-means is weighted by the amount of samples per cluster found, such that a cluster with lots of samples in it will have a bigger impact on the aggregation step compared to a cluster with fewer samples. That is, we modify the k-means objective function (see supplement A for the original) into:

$$F_{km} = \sum_{j=0}^M \min_{C_i \in C_g} (S_j \|C_j - C_i\|^2) \quad (1)$$

where S_j is the amount of samples corresponding to local cluster C_j . Note that the *cluster means* sent back by the clients are at the server used as the *samples* for clustering using k-means. Doing the aggregation with a k-means clustering, we solve the cluster alignment problem, since similar clusters will be close to each other and thus merged by the global k-means step.

Because the amount of samples have to be reported to the central server, there exists a privacy risk if a client finds a cluster with only one sample in it. To prevent this, any clusters holding less than p samples (we used $p = 2$ throughout this work) are simply omitted from the list of means sent to the server.

4 Results

We compared our federated k-means (FKM) with a k-means clustering that is executed on all data centrally, as well as to one-shot the method of Dennis et al. [5]. Our first set of experiments is on simulated data, such that ground truth labels of the cluster centers is known. We therefore calculate the Adjusted Rand Index (ARI) for both central and federated approaches with respect to the labelled samples data. Since there are no labels for the clustering in the FEMNIST experiment, the silhouette score was used instead. In some cases, we added an “informed” setting for Dennis et al., in which we set K_l such as to achieve the highest ARI score by exhaustive search. In all other cases, we run their method using $K_l = K_g$, as the ARI score is only available when ground truth labels are known, which is not always the case.

4.1 Clients Holding Different Parts of the Data

In order to validate the FKM algorithm, a synthetic two-dimensional dataset was generated. The generation procedure is taken from Servetnyk et al. [18]. Sixteen

cluster centers were chosen with an equal distance (here 5) from one another, see Fig. 1a. Then, 50 data points were sampled around each cluster center using a normal distribution (with variance 1). This data was then distributed among four clients in the following way: First, each client is assigned a ‘location’ within the field ($X_1, X_2 \in (-12.5, 12.5)$). From there, the probability P that a data point would be assigned to a certain client scales inversely with the euclidean distance d to that datapoint:

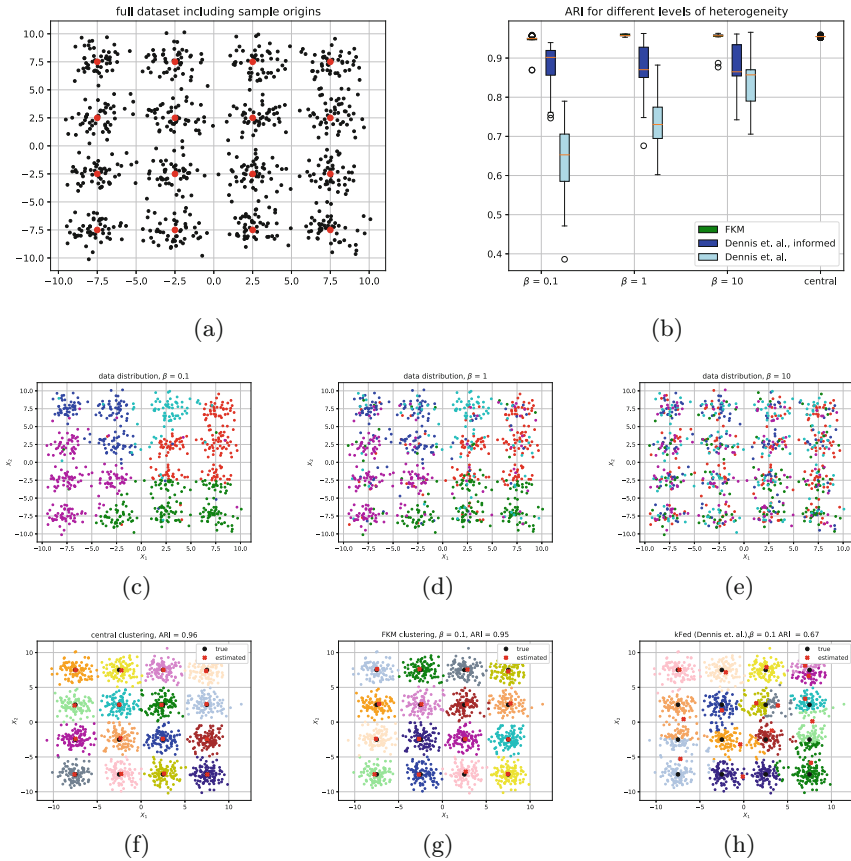


Fig. 1. The regular synthetic datasets. (a) shows the original sampling of the regular synthetic dataset, with the defined cluster means (from which the data are generated using a normal distribution $N(0,1)$) in red. (b) shows ARI results on all three datasets. (c) until (e) shows how this dataset is distributed over five different clients using different values of β . Different colors indicate the different clients. (f) to (h) show examples of a clustering on (c) as given by a centralized k-means, FKM (ours), and kFed (Dennis et al.), respectively. Note that different colors in the last three plots denote different cluster assignments instead of different clients. (Color figure online)

$$P = 1 - \exp\left(-\frac{\beta}{d}\right) \quad (2)$$

where β is a parameter which can be tuned to promote more or less heterogeneity in the data separation. Differing from [18], if a data point happens to be assigned to multiple clients, it instead gets assigned at random.

We wanted to explore the influence of data heterogeneity, i.e. a varying amount of clusters per client. To do so, we generated three versions of this dataset, with $\beta = 0.1, 1, 10$. See Fig. 1 c-e for the final distributions. Note that β only changes which points get assigned to which client, meaning that it does not influence the performance for the central case. Figure 1b shows that our method is able to attain performance similar to a centralized k-means clustering, while outperforming Dennis et al., regardless of tuning of the K_l parameter. Performance of our FKM approach seems to be independent of β (in contrast to the method of Dennis et al.), meaning that our algorithm is robust to having varying cluster amounts between clients.

4.2 Increasing Levels of Noise

Next, we explored the effect of having noisier clusters. We recreated the regular synthetic dataset, but varied the standard deviation from which samples are being generated, from 1 to 1.5 (original used 1). Figure 2 shows the effect. We generated these datasets twice, once with 50 points per cluster and once with 200 points per cluster.

Results on these datasets are shown in Fig. 3. For both central and federated clustering, the ARI scores go down for higher noise levels. This is expected, as

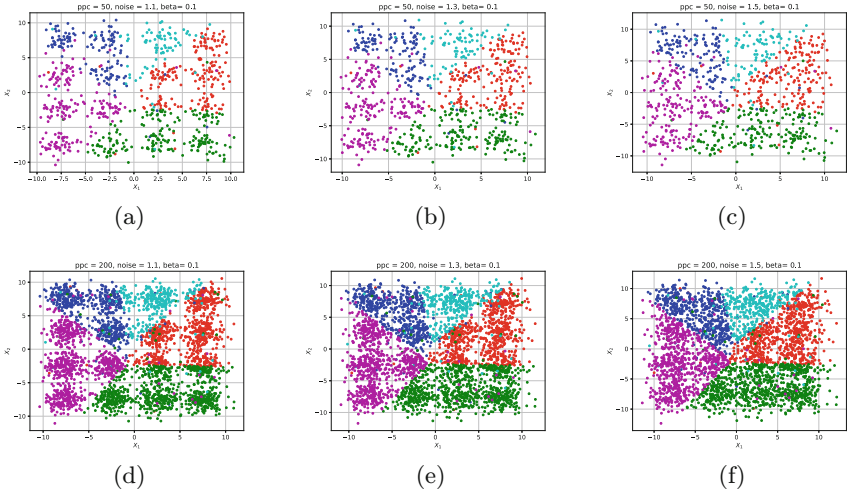


Fig. 2. Some of the data distributions of the simulated datasets with increasing levels of noise (columns), using 50 or 200 points per cluster (rows).

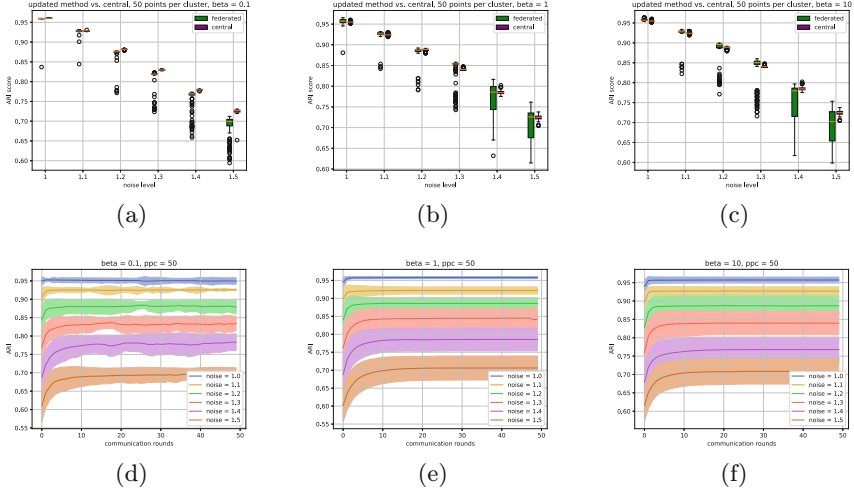


Fig. 3. Clustering results on the synthetic dataset when using different levels of noise for different values of β . (a) to (c) show the final ARI scores for $\beta = 0.1$, 1 and 10, respectively. (d) to (f) show how the ARI score for FKM converges over time, each corresponding to the figure above it.

there will be more points ending up closer to the cluster they did not originally belong to, meaning that even if kmeans finds the original cluster means perfectly, the label assignment will be off. Therefore, the relative difference between federated and central clustering is more important than the absolute ARI scores. Our method attains a similar average performance; however, variance seems to increase compared to centralized clustering. Furthermore, for $\beta = 0.1$, mean ARI decreases compared to central clustering at high noise levels, meaning that a setting with high noise as well as high cluster variability is still a hard challenge for our federated k-means algorithm.

Regardless, performance does seem to increase significantly as compared to the method of Dennis et al. [5]. This can partly be due to our ability to iterate. Figure 3a to 3b shows that, especially for noisier datasets, there is a large benefit in being able to iterate more often. The amount of points per cluster does not seem to influence ARI score significantly, see supplement B.

4.3 Ablation Study

To explore the importance of several parts of our algorithm, we perform an ablation study on the two dimensional synthetic data introduced in Sect. 4.1 with a noise parameter of 1. We make five separate ablations, as well as one setting in which all five modifications are included:

- **Retain empty clusters:** For this ablation, we skip the step where we prune the empty clusters, effectively fixing K_l equal to K_g in all clients.

- **Initialization:** Instead of initializing local cluster using k-means++, we initialize local clusters at random.
- **100 local iterations:** Instead of performing one iteration of k-means locally, we perform 100 iterations.
- **Unweighted aggregation:** When aggregating cluster means on the server, instead of weighting the cluster means by the amount of samples corresponding to said cluster locally, we simply give every cluster a weight of 1 (Note that we explored several values for the amount of local iterations, however we saw little difference between those values, so for sake of brevity we report only the results using the largest value that we explored (100)).

We considered the same values of β as earlier described in Sect. 4.1, i.e. $\beta \in \{0.1, 1, 10\}$, as shown in Fig. 4. This figure shows that the degree to which different parameters are important depend on β , i.e. the heterogeneity of the data. For $\beta = 0.1$ the largest impact can be seen for doing weighted aggregation and local pruning. When data is homogeneously distributed ($\beta = 10$), i.e. every client has data for each cluster, the proposed ablations seem to have little impact on the algorithm. In fact, increasing the amount of local iterations could even be beneficial in a completely heterogeneous case. This is in line with literature on supervised federated learning, where increasing the amount of local epochs can increase performance [15]. However, under heterogeneous circumstances, in supervised federated learning, clients might move too far into local optima before aggregation, decreasing performance with more local iterations [20]. We therefore hypothesize that in even more heterogeneous circumstances, a lower amount of local iterations could still be beneficial for our method as well.

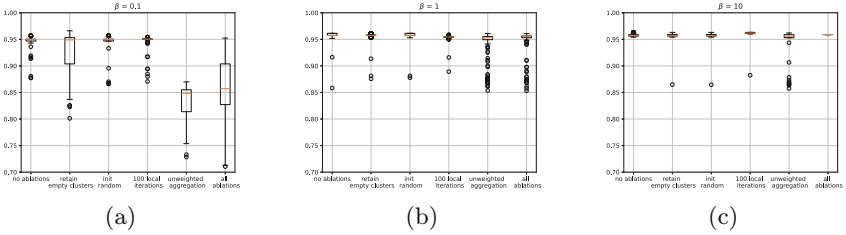


Fig. 4. Results of the ablation study. (a), (b) and (c) show results for $\beta = 0.1, 1$ and 10 , respectively. (a) shows that, for a highly heterogeneous dataset, both dropping empty local clusters and especially weighted aggregation have a large impact on model performance. However, as the data becomes more homogeneously distributed, (subfigures (b)–(c)), these factors become less important.

4.4 High Variability in Number of Local Clusters

Next we wanted to explore the effect of having an even more variable local k . We used the same data as generated for the regular synthetic dataset, but distributed

even more heterogeneously, such that each client only had data from 1, 4, 7, 10 or 16 clusters, respectively. See Fig. 5a.

Figure 5b shows that our method attains a similar average performance as compared to the central case, however with a larger variation. This is probably caused by differences in initializations. If the algorithm initializes in such a way that clients assign data to more clusters than what is being present in their data, the algorithm has a hard time correcting for that. Furthermore, it does not help that one client only has ten datapoints in total, meaning it initializes ten clusters of size one, of which none are being send to the central server due to privacy issues. Regardless, our method does outperform the algorithm from Dennis et al. [5]. This is likely due to our algorithm’s ability to change the value of k for its local k -means step between clients.

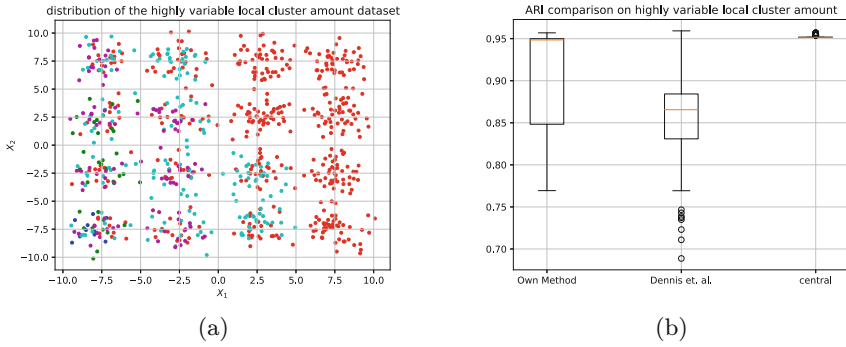


Fig. 5. Assessment of the method on data with a large variability of local clusters per client. (a) shows the distribution per client, (b) the ARI results for different methods.

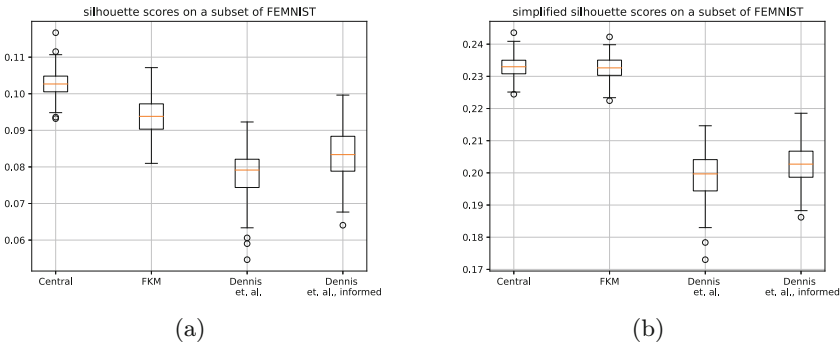


Fig. 6. Results on (a subset of) FEMNIST. (a) shows the silhouette score, (b) the simplified silhouette score.

4.5 Clustering Higher Dimensional Real Data

So far, all our experiments have been done on two dimensional, simulated data. For many use cases, however, data has a much higher dimensionality. In order to determine performance on a higher dimensional dataset, the Federated Extended MNIST (FEMNIST) from LEAF ([3]) (having a dimensionality of 784) was used, which separates the original Extended MNIST ([4]) handwritten numbers and letters based on the person who wrote them. FEMNIST has a dimensionality of 784. This leaves approximately 110 datapoints per client; see supplement C for the distribution. Only 10 clients were used from the original FEMNIST, as this drastically sped up the experiments, while keeping enough data for a meaningful assessment. We set $k = 60$, in line with earlier experiments from Dennis et al. [5]. Figure 6a shows that our method outperforms both settings of the method from Dennis et al. There is still a difference with a central clustering, however. This could be due to the relatively small amount of samples per client compared to the amount of dimensions, decreasing the quality of the local clusters.

The FEMNIST experiments use the silhouette score [16] as their performance metric. The silhouette score involves calculating distances from each point in a dataset to each other point in a dataset. This means that, to calculate a ‘global’ silhouette score, distances between datapoints from different clients need to be determined, something that can not be done in a straightforward federated manner. In our case, the simulated federated environment made it possible to calculate the silhouette score for evaluation purposes. In a real-life setting, the simplified silhouette score ([9]) could be a suitable alternative, as it only calculates distances between datapoints and cluster means, something which can be done on all clients separately.

We compare the simplified silhouette score with the silhouette score from the same experiments in Fig. 6b. There seems to be a high correlation between the two scores for a given method, which is in line with previous work [19].

4.6 Clustering Real Biological Data

Finally, we explore a common clustering task in bioinformatics, that of cell-type identification. We use data from Bouland et al. [2], specifically their dataset referred to as ‘four cancers’. This dataset consists of single-cell RNA sequence measurements from 12 different cancer patients with one of four cancer diagnoses: Lung, endo, colon and renal cancer. In total, there are 22815 genes (features) measured in 132549 cells (samples). These samples are either from tumor tissue or from normal tissue, adjacent to tumour tissue.

Before distributing the dataset over separate clients, we run a standard single-cell pipeline protocol¹ using the Seurat R library [6]. Briefly, we first filter out genes that have less than 200 or more than 2500 feature counts. We then log-normalize the data. Afterwards, we run the Seurat function “FindVariableFeatures” to find the top 500 genes with most variance. Then, we select only the

¹ https://satijalab.org/seurat/articles/pbmc3k_tutorial.html.

samples from tumor tissue, leaving a total of 68905 samples. Finally, we perform dimensionality reduction on these samples using the selected 500 genes, to reduce dimensionality to 5 (determined using the elbow method).

Data is grouped by patient, meaning that all data from a single patient will end up on a single client. Data is being distributed over three clients. Two different distributions of the data are considered, denoted as IID (Identically and Independently Distributed) or non-IID. For the IID-distribution, each client gets data from every cancer type, whereas for the non-IID data, each client gets data from only one cancer type, as well as from two out of six lung patients.

We run FKM, as well as central clustering and kFed (the method by Dennis et al. [5]), with a (global) K of 4, equivalent to the amount of cancers in the dataset. Figure 7 shows the silhouette and simplified silhouette scores for all algorithms. In the IID setting, our method slightly outperforms kFed, while there is still a slight gap with a centralized clustering. However, when considering the non-IID setting, we observe that the performance gap between FKM and kFed increases, whereas the difference between FKM and a centralized clustering is similar compared to the IID setting, indicating that FKM is more robust to various data distributions.

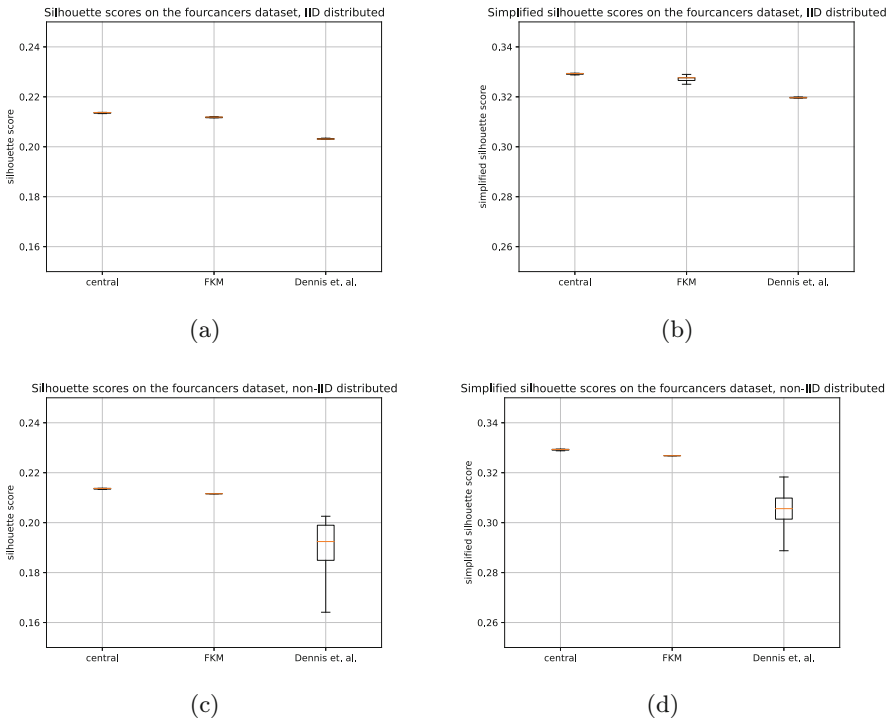


Fig. 7. Silhouette and simplified silhouette scores on the fourcancers dataset. Both the silhouette score in (a) as well as the simplified silhouette score in (b) show increased performance of FKM compared to kFed, though still slightly underperforming a centralized clustering. However, when the data is distributed non-IID (figure (c) and (d)), the gap between the federated methods seems to increase.

5 Discussion and Conclusion

This work describes the implementation and validation of a federated k-means clustering algorithm (FKM), enabling clustering over multiple datasets without sharing the underlying data. Our results show performances close to a central method, in which all data is brought into a single location. There are still some scenarios in which our method shows larger variability in performance as compared to a central clustering, however. These are mostly the more difficult scenarios, such as when there is an extreme distribution in the amount of cluster present on each client, or when the data has a high dimensionality as with the FEMNIST experiment. Assessment of our method on more heterogeneous and 'real life' datasets is therefore an important direction for future work. Nevertheless, FKM has shown to be a promising method in finding similarities among distributed datasets without the need of sharing any data.

6 Code Availability

The code to run FKM, as well as all experiments and generate the figures used throughout this manuscript, can be found at: <https://github.com/swiergarst/fedKMeans/>.

A Background on K-Means and K-Means++

A.1 K-Means Clustering

The objective of a clustering algorithm is to partition a given dataset into several subsets with similar features. The k-means clustering algorithm does so by trying to minimize the within cluster sum-of-squares criterion:

$$F_{km} = \sum_{j=0}^m \min_{C_i \in C} (||X_j - C_i||^2) \quad (3)$$

with m the amount of samples, C_i the cluster mean of cluster i , C the set of all cluster means and $X_{j,k}$ being data point j assigned to cluster k . The procedure in which the k-means algorithm tries to minimize Eq. 3 consists of two steps. First, all data points get assigned to the cluster mean according to the lowest euclidean distance. Then, the mean center point from all points assigned to a certain cluster is calculated. This is done for every cluster, creating a new set of means to start the next round with. This process is repeated until the change within these means is smaller than a certain threshold (and the algorithm has reached convergence) ([7]).

A.2 K-Means++

One of the drawbacks of classical k-means clustering is that its initialization is sampled uniformly from the underlying data. This means that having initial

cluster means that all come from the same cluster is as probable as having initial cluster means spread across all clusters. Although the K-means algorithm itself can somewhat compensate for this, it still leads to large variability in performance. Arthur and Vassilvitskii developed an initialization method for K-Means to combat this high variability, called k-means++ [1]. Instead of sampling K cluster means from the data with uniform probability, datapoints get weighted based on their distance to the closest already mean that is chosen, with larger distances giving larger weights. This results in (on average) initializations that are more distributed over the space, and prevents (on average) initial cluster means from starting very close to each other, decreasing k-means performance.

B Extra Results on Increasing Amount of Points per Cluster

See Fig. 8.

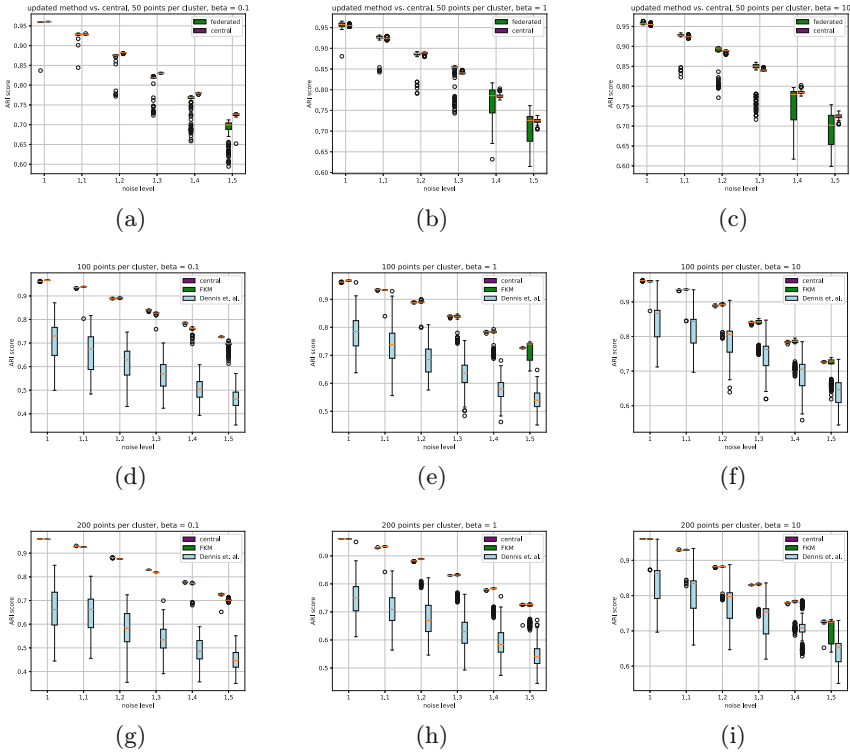


Fig. 8. results on using different levels of noise for different values of β , with differing amounts of points per cluster. From left to right, the columns correspond to $\beta = 0.1$, 1 and 10 respectively. From top to bottom, the rows correspond to 50, 100, and 200 points per cluster.

C FEMNIST Distribution

See Fig. 9.

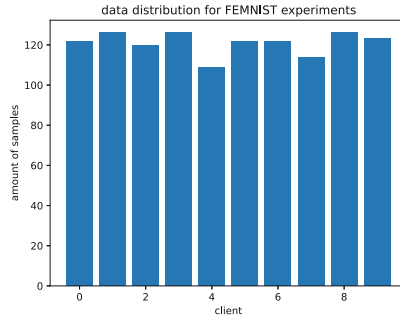


Fig. 9. Sample distribution for the FEMNIST dataset

References

1. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. Technical Report 2006-13, Stanford InfoLab (2006). <http://ilpubs.stanford.edu:8090/778/>
2. Bouland, G.A., Mahfouz, A., Reinders, M.J.T.: Differential analysis of binarized single-cell RNA sequencing data captures biological variation. *NAR Genom. Bioinform.* **3**(4), lqab118 (2021). <https://doi.org/10.1093/nargab/lqab118>
3. Caldas, S., et al.: LEAF: a benchmark for federated settings. *CoRR* abs/1812.01097 (2018). <http://arxiv.org/abs/1812.01097>
4. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: an extension of MNIST to handwritten letters. *CoRR* abs/1702.05373 (2017). <http://arxiv.org/abs/1702.05373>
5. Dennis, D.K., Li, T., Smith, V.: Heterogeneity for the win: one-shot federated clustering. In: Meila, M., Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 139, pp. 2611–2620. PMLR (2021). <https://proceedings.mlr.press/v139/dennis21a.html>
6. Hao, Y., et al.: Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nat. Biotechnol.* (2023). <https://doi.org/10.1038/s41587-023-01767-y>
7. Hartigan, J., Wong, M.: Algorithm as 136: a k-means clustering algorithm. *J. Roy. Stat. Soc. Ser. C (Appl. Stat.)* **28**, 100–108 (1979)
8. Hou, R., Tang, F., Liang, S., Ling, G.: Multi-party verifiable privacy-preserving federated k-means clustering in outsourced environment. *Secur. Commun. Networks* **2021**(1), 3630312 (2021)
9. Hruschka, E., de Castro, L., Campello, R.: Evolutionary algorithms for clustering gene-expression data. In: *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pp. 403–406 (2004). <https://doi.org/10.1109/ICDM.2004.10073>

10. Kumar, H.H., Karthik, V.R., Nair, M.K.: Federated K-means clustering: a novel edge AI based approach for privacy preservation; federated k-means clustering: a novel edge AI based approach for privacy preservation. In: 2020 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM) (2020). <https://doi.org/10.1109/CCEM50674.2020.00021>
11. Lee, G.H., Shin, S.Y.: Federated learning on clinical benchmark data: performance assessment. *J. Med. Internet Res.* **22**(10) (2020). <https://doi.org/10.2196/20891>
12. Li, Q., et al.: A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Tran. Knowl. Data Eng.* 1–1 (2021). <https://doi.org/10.1109/TKDE.2021.3124599>
13. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: challenges, methods, and future directions. *IEEE Signal Process. Mag.* **37**(3), 50–60 (2020)
14. Liu, Y., Ma, Z., Yan, Z., Wang, Z., Liu, X., Ma, J.: Privacy-preserving federated k-means for proactive caching in next generation cellular networks. *Inf. Sci.* **521**, 14–31 (2020). <https://doi.org/10.1016/J.INS.2020.02.042>
15. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.V.: Communication-efficient learning of deep networks from decentralized data. In: Singh, A., Zhu, J. (eds.) *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 54, pp. 1273–1282. PMLR (2017). <https://proceedings.mlr.press/v54/mcmahan17a.html>
16. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987). [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7). <https://www.sciencedirect.com/science/article/pii/S0377042787901257>
17. Sadilek, A., et al.: Privacy-first health research with federated learning. *NPJ Digit. medicine* **4**(1), 132 (2021). <https://doi.org/10.1038/s41746-021-00489-2>, <http://www.ncbi.nlm.nih.gov/pubmed/34493770>, <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC8423792>
18. Servetnyk, M., Fung, C.C., Han, Z.: Unsupervised federated learning for unbalanced data. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference* (2020). <https://doi.org/10.1109/GLOBECOM42002.2020.9348203>
19. Wang, F., Franco-Penya, H.-H., Kelleher, J.D., Pugh, J., Ross, R.: An analysis of the application of simplified silhouette to the evaluation of k -means clustering validity. In: Perner, P. (ed.) *MLDM 2017. LNCS (LNAI)*, vol. 10358, pp. 291–305. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62416-7_21
20. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. *CoRR abs/1806.00582* (2018). <http://arxiv.org/abs/1806.00582>



Feature Selection Voting Strategies and Hyperparameter Tuning in a Boosting Classification

Nicole Dalia Cilia^(✉), Giovanni Fanara, and Vincenzo Conti

Department of Engineering and Architecture, University of Enna Kore, Enna, Italy
nicoledalia.cilia@unikore.it

Abstract. This paper presents a novel approach to feature selection voting strategies and a combination of hyperparameter optimization techniques to improve the performance of boosting classification. As a case study, the paper focuses on developing robust predictive models capable of accurately classifying different degrees of damage in concrete structures. By leveraging boosting algorithms and optimization strategies, the proposed methodology aims to enhance the efficiency and accuracy of damage classification processes. The findings contribute to advancing the field of structural health monitoring and maintenance by providing efficient and reliable methods for assessing damage in reinforced concrete structures. Experimental results demonstrate the effectiveness of the approach in accurately identifying damage levels in reinforced concrete frames. Moreover, this work improves the performance of the boosting classification models and identifies the most relevant sensors. After the feature selection process and hyperparameter optimization, the best experimental result reaches an F1-score of 0.919, identifying as best accelerometers those located closer to the ground.

Keywords: Boosting techniques · Machine learning · Optimization strategies · Predictive modeling · Feature selection · Damage classification

1 Introduction

In recent years, machine learning (ML) techniques have become instrumental in enhancing the precision and effectiveness of structural health monitoring. One of the most promising and impactful applications of ML lies in the realm of civil infrastructure, where the intersection of cutting-edge technology and the built environment has the potential to reshape the future of urban development. The accelerating pace of urbanization and the increasing complexity of modern cities have heightened the demand for resilient and efficient civil infrastructure systems. Machine learning, with its ability to analyze vast datasets, derive meaningful insights, and optimize processes, emerges as a transformative force capable of addressing these challenges head-on.

In the realm of structural engineering, the evaluation and optimization of frames play a pivotal role in ensuring the safety, resilience, and longevity of critical infrastructure. With the ever-increasing complexity of modern buildings and the imperative to enhance our ability to respond to unforeseen events, there is a growing demand for advanced methodologies that can accurately classify and optimize the assessment of damage in reinforced concrete structures.

This paper proposes a robust approach that combines boosting classification techniques with optimization methods to offer a comprehensive solution for the efficient and accurate evaluation of damage in frames. As the consequences of structural failures can be catastrophic, the development and implementation of robust methodologies are essential to prevent and mitigate potential risks. Boosting algorithms, known for their ability to enhance the performance of classification models, are employed to effectively identify and categorize damage patterns in reinforced concrete frames. These algorithms leverage the strengths of multiple weak classifiers, collectively improving the overall accuracy and reliability of the assessment process. The integration of optimization techniques further refines the classification results, allowing for the hyperparameter fine-tuning to achieve an optimal balance between accuracy and computational efficiency.

Throughout this paper, we will delve into the theoretical underpinnings of the proposed boosting classification and optimization approach. Additionally, a practical application and case study will be explored to demonstrate the effectiveness of this methodology in a real-world scenario. We propose a pipeline for an application in the civil engineering field, specifying which types of sensors can be used and which ML techniques apply. This is the first step for a work whose complement could be to validate the experimental set in a real scenario, replicating the study with satisfactory results using another structure with similar characteristics. By presenting a comprehensive overview of the advantages of our approach, we aim to contribute to the advancement of state-of-the-art methodologies for damage assessment in frames.

One of the most important and delicate phases of this work is the selection of characteristics. The feature selection process aims to identify the features that contribute most significantly to discriminating among the classes of interest, thereby enhancing the system's explanatory power. Additionally, as another objective of this study was to ascertain the relative importance of different sensors, we conducted a guided feature selection for groups of features representing information derived from individual sensors. We implement some known literature selection techniques and the results obtained by each of them have been recorded as votes for each feature.

The structure of this paper is organized as follows: In Sect. 2, we provide a comprehensive review of related work, highlighting recent advances in the field of artificial intelligence applied to damage assessment in reinforced concrete structures. This serves to contextualize our study within the current research landscape. Section 3 details the dataset used for validating our approach, including the types of sensors employed and the nature of the data collected. Following this, in Sect. 4, we discuss the data preprocessing phase, which is crucial for ensuring

the accuracy of machine learning models. Section 5 is devoted to the methodology adopted, illustrating the boosting techniques applied and the parameter optimization strategy to further enhance model performance. In Sect. 6, we present our findings, showcasing the effectiveness of the proposed approach through a comparative analysis of the performance of different models. Finally, in Sect. 7, we conclude our study, reflecting on the contributions made and suggest future directions for research in this area. Through this work, we aim to provide a solid foundation for further developments in the field of structural damage assessment, proposing innovative solutions to address present and future challenges in the monitoring of the health of reinforced concrete structures.

2 Related Works

In the last few years, ML based tools have demonstrated their ability to solve a wide spectrum of real-world problems of civil engineering, from innovative machine learning algorithms enhancing structural performance predictions to the integration of intelligent systems in the construction process. The intersection of ML and structural civil engineering has emerged as a dynamic and transformative field, revolutionizing traditional approaches to design, analysis, and optimization of civil infrastructure.

Based on the literature reviewed, we can categorize the studies in the field into the following thematic groups:

Machine Learning for Structural Health Monitoring (SHM). Sharma et al. in [24] focus on using a hybrid AI approach that combines machine learning and deep learning for structural monitoring and health assessment of civil infrastructure. Liu in [21] discusses the growing interest in ML for SHM, with a specific focus on the application of wireless sensor networks (WSNs) and machine learning algorithms. The authors stress the importance of WSNs that, thanks to their ability for remote and continuous data acquisition, are replacing traditional wired systems, allowing more efficient monitoring and lower costs. Lee et al. in [20], relating to the analysis of changes in the mechanical relationships established for the target structure, present a novel structural damage detection methodology that utilizes AI networks for identifying damage location and extent based on a reference model. In Eltouny et al. [11] a Bayesian-optimized unsupervised learning technique to detect structural damage is presented.

Machine Learning Algorithms for Damage Prediction and Analysis. Lazaridis et al. in [19] offer a comparative analysis of ten machine learning algorithms' performance in predicting seismic damage in reinforced concrete structures. Figueiredo and Santos in [12] examine the use of various machine learning algorithms for damage detection in civil, mechanical, and aerospace structures, highlighting the importance of learning structural behavior from experience. Kostinakis et al. in [18] presents a method for evaluation the seismic damage potential in buildings. This study employs machine learning algorithms and a large training dataset to predict seismic performance efficiently, and it also distinguishes different classes of damage levels. The study presented Nonlinear Time

History Analysis for 90 three-dimensional reinforced concrete buildings with varied number of stories and bays, by using available 65 real seismic records. Bhatta et al. in [6] conducted research oriented towards implementation of machine learning methods to forecast the seismic behavior of different structural system. In this paper, the authors have used various machine learning algorithms, e.g., support vector machines and neural networks, to study geological attributes in seismic exploration. Athanasiou et al. in [3] proposes an application of different machine learning models in the prediction process for seismic demand imposed on building structures. Several algorithms are used by the authors to gather insights on how they can lead to better predictive accuracy of seismic performance, it could be any algorithm such as decision trees and random forests.

Damage Localization and Quantification Techniques. Dhiraj et al. in [1] proposed the use of the monitored structure's transmissibility functions for the structure under observation, which can be fed into a novel composite architecture consisting of a Deep CNN followed by multivariate linear regressors to detect, localize, and quantify the damage extent in a system. The paper aims to improve resource utilization in terms of structure maintenance and its longevity. Byung Kwan Oh et al. in [22] studied and proposed an unsupervised damage identification method using a convolutional neural network (CNN) trained exclusively with healthy state data. The discrepancy between healthy state data and output data from the CNN with the damaged state response was displayed as damage indicators.

Automated Structural Damage Detection (SDD) in Extreme Events. Bai et al. in [4] focus on automated SDD using deep learning methods for processing 2D images, particularly in the context of extreme events like large earthquakes. This includes the use of ResNet for classifying multiple classes in SDD tasks and the development of end-to-end networks for directly detecting cracks and spalling. Compared to the presented studies, this article sets itself apart by employing optimized boosting models developed with data from both triaxial and monoaxial piezoelectric accelerometers, aiming not only to classify damage accurately but also to pinpoint the most relevant sensors, thereby enhancing the precision of damage assessment in reinforced concrete frames. This focus on sensor relevance and optimization distinguishes our study from existing literature, offering new insights into the potential of machine learning.

3 Dataset Description

The tests conducted in this work utilize the broadband random noise tests (BN) which is a testing method used to assess the dynamic response of structures subjected to wideband random vibrations. It is used for simulating the real environmental forces that a structure might encounter during use, such as vibrations caused by traffic or industrial activities. In this work, these tests were conducted on reinforced concrete frames, applying vibrations at varying intensities from 0.1 m/s^2 to 2.0 m/s^2 and different durations to identify and analyse the damage

phases of the structure under various stress conditions. Data provided in the form of 108 files with a .DAT extension contains features related to the input of the vibrating platform, such as acceleration, displacements, and velocities in the X, Y, and Z directions, as well as the outputs measured by the sensors, including the name of the measurement channel (e.g., A1X), and the direction of measurement. Each file contains measurements spanning 50 s. The provided data were collected ergodically, and at the beginning and end of each data set, there are amplitude modulations, which can be either ascending or descending.

4 Preprocessing Phase

A preprocessing phase is needed to train machine learning algorithms. The original dataset consisted of 108 files, each with 50,000 rows and 35 columns. Initially, a column cleaning phase was carried out in which the columns not useful for the project were removed. At the end of this phase, a total of 13 columns remained, representing the sensors. Consequently, the dataset was resized by first applying a fragmentation into 5 parts, resulting in a total of 540 different files, each with 10,000 rows and 13 columns. Four statistical techniques (median, minimum value, maximum value, and standard deviation) were then applied to the columns of each file to reduce each CSV file to a single sample. By concatenating all the samples, a single dataframe of 540 rows and 52 columns was obtained. Finally, the column corresponding to the label was added, resulting in a total of 53 columns. The dataset was explored to analyze the distribution of the classes. Each class represents a specific degree of damage suffered by the wall. Each particular stress corresponds to a certain degree of damage, indicated by a particular color. In total, we can identify 12 different degrees of damage, which differ in the intensity and time history of the applied stress. From Fig. 1, it is noteworthy that the distribution of samples in the 12 classes to be predicted is not balanced. Some classes, like the “yellow” class, include almost a quarter of

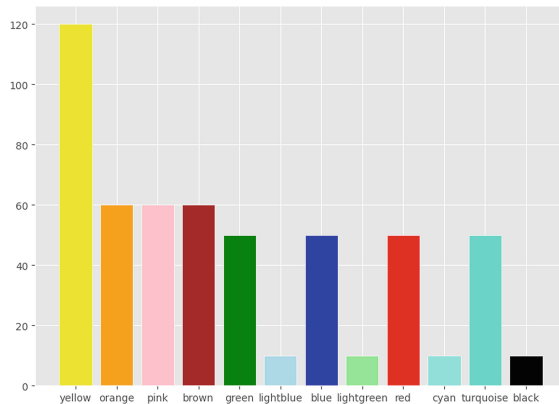


Fig. 1. Distribution of samples in classes. (Color figure online)

the entire number of samples, while others, like the “black” class, contain only 10 samples, less than 2% of the dataset.

After encoding the structural damage levels into numerical labels ranging from 1 to 12, the dataset was divided into training and test sets. To mitigate the issue of an unbalanced dataset, which could lead to under-represented classes in the train and test sets, a stratified split approach was employed. The respective percentages of 70% and 30% were chosen for splitting data into training and test sets, with the samples being mixed before proceeding with the split.

5 Benchmark Methodology

To train four different boosting algorithms to predict the degree of damage to the structure across 12 classes, we employed two hyperparameter optimization techniques and four types of feature selection methods. The performance of these models was assessed using ten distinct metrics.

The study was conducted on a laptop equipped with the following hardware components:

- CPU AMD Ryzen 7 3700U;
- GPU AMD Radeon RX Vega 10;
- 16 GB RAM;
- 512 GB SSD.

The laptop ran Ubuntu 22.04.3 LTS as the operating system. All code executions were performed in Visual Studio Code, utilizing a Conda environment with Python version 3.11.5. All libraries used were updated to their latest versions as of the publication date of this paper. Figure 2 illustrates the experimental pipeline. In particular, the process begins with the preprocessing phase. Subsequently, hyperparameter optimization of the boosting models used in the first classification experiment is performed. After obtaining the best models through hyperparameter optimization, the most relevant features are selected. This process aims to identify the features that contribute most to distinguishing the

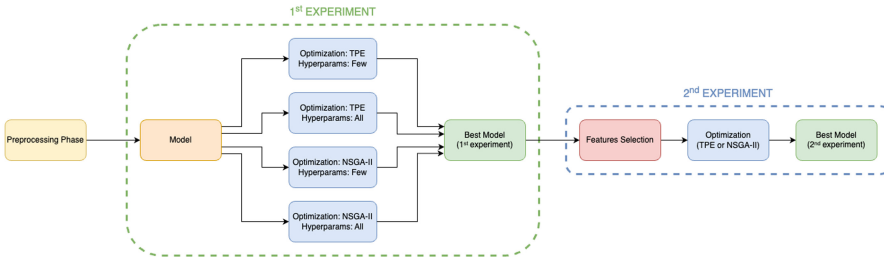


Fig. 2. Steps of the proposed experiments pipeline: preprocessing phase; hyperparameters optimization (first experiment); feature selection and hyperparameters optimization (second experiment). The second experiment trains the best models obtained from the first experiment using a feature selection and a new hyperparameter optimization.

classes of interest. Once the most informative features are selected, the best models undergo further optimization to maximize classification performance. It is noteworthy that this workflow is repeated for four different boosting models to evaluate their performance and compare their effectiveness. In the following details of the methodological steps utilized are presented.

5.1 Hyperparameters Tuning

In this study, two methods of hyperparameter optimization were used:

- Non-dominated Sorting Genetic Algorithm II (NSGA-II);
- Tree-structured Parzen Estimator (TPE).

Non-dominated Sorting Genetic Algorithm II (NSGA-II). NSGA-II [9, 10] is a multi-objective genetic algorithm commonly used to solve optimization problems with multiple simultaneous objectives. However, it can be adapted to address problems with a single objective.

Tree-Structured Parzen Estimator (TPE). The Tree-structured Parzen Estimator (TPE) is a hyperparameter optimization technique that leverages conditional probability density models to effectively explore hyperparameter space [5, 27].

Both techniques were implemented using the Python Optuna library. Since our study performs a multi-class classification on an imbalanced dataset, the search for the best hyperparameter values and model training focused on getting the highest possible value of the F1-score macro metric [2]. The hyperparameter search was carried out by training the model using a 5-Fold Cross-Validation with a stratified division of the training set [25, 26].

5.2 Metrics

The search for the best hyperparameter values and training of the models focused on achieving the highest possible value of the F1-macro metric. Since the dataset is unbalanced, the F1-macro gives equal importance to all classes, ensuring that the performance is distributed across all represented classes, thus enhancing the model's robustness. Other metrics have been calculated, as reported in Tables 2 and 3, to show the complete performance of the model: accuracy, precision, sensitivity, specificity, G-mean (The Geometric Mean given by the square root of the product of Sensitivity and Specificity) and ROC AUC. The use of multiple evaluation metrics is crucial to obtaining a complete and accurate understanding of the model's performance. Each metric provides a different and complementary perspective on the model's strengths and weaknesses, allowing us to make a more informed choice.

5.3 Feature Selection

The feature selection process aims to identify the features that contribute most to distinguishing the classes of interest, improving the system's explanatory power. Furthermore, since another goal of this study was to determine which sensors play a more significant role, we performed guided feature selection for groups of features representing information derived from a single sensor.

We implement the following feature selection techniques:

- Model feature importance [23];
- Analysis of Variance (ANOVA) [15];
- Kendall's Score [17];
- Recursive Feature Elimination (RFE) [8].

Only the last one is a wrapper method that requires training the classification algorithms.

For each technique, 35 features were selected because it was expected that, following the voting policy implemented in this work, fewer features would remain. In fact, the number 35 allows to have in the end a quantity of selected features equal to 30–60% of the initial features.

For the Feature Importance techniques, we selected features whose scores exceeded the threshold of 0.02. This threshold was chosen because the method of feature importance needed to select about 30/35 features out of the total 52, similar to the other three feature selection techniques.

The results obtained by each feature selection method were recorded as votes for each feature. If a feature was selected by a method, it received a vote of 1; otherwise, it received a vote of 0. Later, a majority vote calculation was made for each feature, considering a threshold, to determine which features were selected by most methods. Since the number of techniques implemented is even, a feature was selected only if it received at least 3 votes out of 4.

The dataset features originated from seven monoaxial and triaxial high-sensitivity piezoelectric accelerometers. For sensors equipped with three axial components, three distinct features were generated, whereas monoaxial sensors contributed a single feature each. During the feature selection process, there exists a possibility that features corresponding to one or more axial components of the triaxial sensors may be eliminated. Retaining an incomplete set of features from any of these triaxial sensors would be inconsistent with identifying the most effective sensors. To prevent partial feature selection for these sensors, we implemented the following strategy:

- If one in three features is excluded, it is retained along with the other two previously selected features;
- If two out of three features are excluded, the remaining feature that was previously selected is also excluded;
- If all three features are retained or excluded, no further action is taken.

Initially, the dataset comprised 13 accelerometer features. However, through preprocessing, this number increased to 52. This resulted from computing four

statistical indices for each original feature, necessitating controls for the partial selection of a sensor’s features to be applied fourfold. Subsequently, for each statistical index calculated, we examined the features of the three components of each sensor, ensuring a comprehensive analysis of the data derived from all sensors.

5.4 Model Training

To achieve more robust performance, we employed boosting techniques instead of relying on weak machine learning algorithms. Boosting enhances model accuracy by aggregating the predictions of multiple weak learners, thus reducing both bias and variance. Most boosting models share the same operating mechanics: they build a predictive model by sequentially combining several weak classifiers to create a strong and complex model. This approach not only improves prediction strength but also offers advantages such as increased model interpretability and adaptability to different data distributions. We implemented the following classifiers:

- Adaptive Boosting (AdaB): AdaB [13], with the Decision Tree as the weak classifier, has been implemented through the Scikit-Learn library.
- Gradient Boosting Machine (GBM): Unlike Adaptive Boosting, Gradient Boosting [14] does not adjust the weights of the samples but instead tries to fit the negative gradient of the loss function used in the model. In other words, at each iteration, a new model is added that predicts the residual or error of the overall model constructed up to that point. It can be used with various loss functions but is more complex to optimize compared to AdaB since it requires the configuration of parameters such as the learning rate.
- eXtreme Gradient Boosting (XGB): XGB, short for “extreme Gradient Boosting”, is an extremely efficient and enhanced version of gradient boosting that uses gradient-boosting trees for classification [7].
- Light Gradient Boosting Machine (LGBM): Since we have a fairly large dataset, the last classifier we decided to include in our experiments is LGBM. LGBM uses a leaf-wise growth strategy in which it selects leaves with maximum loss gain for expansion. In addition, the “hist” approach in LGBM refers to how histograms are constructed to accelerate the split-finding process during the tree training phase [16]. LGBM is currently the most versatile type of boosting since it provides extremely high performance in modeling the classifier you want to generate.

Each type of boosting has different hyperparameters to configure. We define two groups of hyperparameters to be optimized for each boosting algorithm: one group with only the most common hyperparameters to be optimized (FEW), and another with all optimizable hyperparameters (ALL).

The hyperparameters of each category for each type of model were optimized using the two methods described above (TPE and NSGA-II), creating a further subdivision into two additional model types. For each model, hyperparameter

optimization was performed twice in a row with 50 trials each, ensuring that the best hyperparameter values obtained from the first search were used as a starting point for the second. This strategy allowed us not only to obtain better results but also to adjust the hyperparameter search space in the second search according to the best values found in the first. For each model category, the hyperparameter value search spaces for both the first and second searches were the same for both types of optimization implemented. After finding the best hyperparameter values, for each model, a 5-Fold Cross-Validation was performed 10 times.

In summary, for each type of boosting, we performed:

- Model with few parameters optimized with TPE;
- Model with few parameters optimized with NSGA-II;
- Model with all parameters optimized with TPE;
- Model with all parameters optimized with NSGA-II.

After training and evaluating the models of each type, a comparison was made to select the model with the best accuracy. Using this model, we performed the feature selection described in Sect. 5.3.

The best model is optimized on the dataset having the features selected by the TPE or NSGA-II, depending on the initially applied method. In this case, hyperparameter optimization is carried out only once with 100 trials.

After the optimization, the model is trained and evaluated, ending with a final comparison with the following models:

- Best model with the dataset having all features;
- Best model optimized with dataset having selected features.

These steps were followed for all types of boosting models, with a final comparison of the best models for each type. Table 1 shows the search spaces of the hyperparameters values for first and second optimization, with few and all features, for each classifier.

6 Results

This section presents in detail the results of two methodologies: in the first experiment we train and test the four models with hyperparameter optimizations; in the second experiment, we introduce the feature selection technique to obtain information about sensors involved in the acquisition data.

6.1 Results with Hyperparameter Optimization

Table 2 shows the performance of the four models (first column: AdaB, GBC, XGB, LGBM), with the two techniques of hyperparameter optimizations (second column: TPE and NSGA), for two subsets of hyperparameters (third column: FEW and ALL). The AdaB models show excellent results. As regards the

Table 1. Hypermarameter optimization

Classif	Hyperp	Few 1st Opt	Few 2nd Opt	All 1st Opt	All 2nd Opt
AdaB	criterion	['gini', 'entropy']	['gini', 'gini']	['gini', 'entropy']	['gini', 'gini']
	max_depth	(5, 15)	(7, 15)	(5, 15)	(7, 15)
	min_samples_split	(2, 20)	(6, 20)	(2, 20)	(6, 20)
	min_samples_leaf	(1, 10)	(1, 12)	(1, 10)	(1, 12)
	minWeightFractLeaf	–	–	(1e–5, 0.5)	(1e–5, 0.1)
	max_features	–	–	['sqrt', 'log2']	['sqrt', 'log2']
	max_leaf_nodes	(5, 50)	(15, 60)	(5, 50)	(15, 60)
	min_imp_dec	–	–	(1e–5, 0.1)	(1e–5, 0.1)
	ccp_alpha	–	–	(1e–8, 0.1)	(1e–8, 0.1)
	n_estimators	(50, 300)	(100, 300)	(50, 300)	(100, 300)
	learning_rate	(0.01, 1.0)	(0.1, 1.0)	(0.01, 1.0)	(0.1, 1.0)
GBC	learning_rate	(0.001, 1.0)	(0.01, 0.3)	(0.0001, 1.0)	(0.001, 0.3)
	n_estimators	(50, 600)	(200, 700)	(50, 600)	(200, 800)
	subsample	–	–	(0.6, 1.0)	(0.8, 1.0)
	min_samples_split	(2, 30)	(4, 28)	(2, 30)	(6, 30)
	min_samples_leaf	(1, 25)	(2, 18)	(1, 20)	(2, 18)
	minWeightFractLeaf	–	–	(1e–5, 0.5)	(1e–5, 0.5)
	max_depth	(3, 15)	(5, 12)	(3, 15)	(3, 15)
	min_imp_dec	–	–	(1e–6, 0.1)	(1e–5, 0.1)
	max_features	–	–	['sqrt', 'log2']	['sqrt', 'log2']
	max_leaf_nodes	(2, 60)	(5, 50)	(2, 50)	(15, 55)
	ccp_alpha	–	–	(1e–8, 0.1)	(1e–8, 0.01)
XGB	n_estimators	(50, 700)	(100, 800)	(50, 700)	(100, 800)
	max_depth	(3, 12)	(3, 10)	(3, 12)	(3, 10)
	learning_rate	(0.001, 0.3)	(0.01, 0.5)	(0.001, 0.3)	(0.01, 0.5)
	tree_method	–	–	['ex', 'appr', 'hist']	['ex', 'appr', 'hist']
	gamma	(1e–4, 10)	(1e–5, 1)	(1e–4, 10)	(1e–5, 1)
	min_child_weight	(0.01, 5)	(0.001, 1)	(0.01, 5)	(0.001, 1)
	max_delta_step	–	–	(1e–4, 10)	(1e–6, 5)
	subsample	(0.6, 1.0)	(0.7, 0.9)	(0.6, 1.0)	(0.7, 0.9)
	colsample_bytree	(0.6, 1.0)	(0.5, 0.8)	(0.6, 1.0)	(0.5, 0.8)
	colsample_bylevel	–	–	(0.6, 1.0)	(0.5, 0.8)
	colsample_bynode	–	–	(0.6, 1.0)	(0.5, 0.8)
	reg_alpha	–	–	(1e–6, 1)	(1e–6, 0.01)
	reg_lambda	–	–	(1e–6, 1)	(1e–4, 100)
LGBM	num_leaves	–	–	(20, 200)	(40, 240)
	max_depth	(3, 11)	(3, 9)	(3, 12)	(3, 10)
	learning_rate	(0.001, 0.1)	(0.01, 0.4)	(0.01, 0.5)	(0.1, 0.8)
	n_estimators	(50, 400)	(100, 300)	(50, 500)	(100, 500)
	subsample_for_bin	–	–	(40, 80)	(40, 80)
	min_split_gain	–	–	(0.001, 1)	(0.001, 0.1)
	min_child_weight	(0.01, 5)	(0.001, 0.1)	(0.01, 5)	(0.001, 0.1)
	min_child_samples	–	–	(1, 60)	(10, 60)
	subsample	(0.6, 1.0)	(0.7, 0.9)	(0.6, 1.0)	(0.8, 1.0)
	subsample_freq	–	–	(1, 50)	(10, 40)
	colsample_bytree	(0.6, 1.0)	(0.5, 0.7)	(0.6, 1.0)	(0.7, 1.0)
	colsample_bynode	–	–	(0.6, 1.0)	(0.5, 0.8)
	max_delta_step	–	–	(0.01, 10)	(0.1, 10)
	reg_alpha	–	–	(0.0001, 1)	(0.01, 1)
	reg_lambda	–	–	(0.01, 100)	(1, 150)
	max_bin	–	–	(100, 300)	(150, 300)
	min_data_in_bin	–	–	(1, 20)	(1, 10)

optimization techniques used, the TPE achieves always the best results. The performance with optimization of all hyperparameters is often better.

Comparing the results obtained from the AdaB models, it emerges that the optimized model with TPE and all possible hyperparameters configured is the best model. The best combination of hyperparameters found for the best model is shown in Table of the Appendix, first row.

GBC, compared to all other types of boosting implemented, presented the best performance. Analyzing the two optimization techniques used, the models optimized with NSGA-II produced better results in terms of F1-score performance. On the other hand, models optimized with TPE performed better in metrics that describe the robustness of a model. In addition, it is noteworthy that models optimized with TPE achieved very similar performance. s for the two categories of models with different numbers of hyperparameters configured, it can be noted that the category with more hyperparameters showed better results.

The best combination of hyperparameters found for the best model, optimized with NSGA-II and having all the hyperparameters configured is in the second row of Table in Appendix.

As can be seen in Table 2 XGB models performed below average. Compared to other types of boosting, XGB has some of the lowest F1-score. However, considering other more specific metrics that describe performance related to each class, it can be noted that the XGB models are as efficient as the others. The hyperparameter optimization has not affected the final performances of the models. The results obtained are similar to each other.

The best model for this type of boosting was the one optimized with TPE and having all the possible hyperparameters configured. The best combination of hyperparameters for the best model is shown in the third row of Table in Appendix.

The results obtained from the LGBM models were not the best. The optimized models with NSGA-II have significantly higher values for this metric than the counterparty and the values obtained for all metrics from models with more optimized hyperparameters are greater than models with less optimized hyperparameters. Thus the wide range of hyperparameters offered by LGBM strongly influence the final performance of the model.

The best combination of hyperparameters found for the best model is presented in Table of Appendix, fourth row.

6.2 Results with Feature Selection

The feature selection conducted on the four types of boosting implemented led to the generation of 4 different datasets with most of the different features. In addition, for each type of boosting, the best model has been trained, tested and optimized on the corresponding dataset having the selected features.

From the results obtained in the four best models, Table 3, emerges that the best performances derive from LGBM. In particular, the F1-score of 0.919 stands out from the other models, achieving performance comparable to the first experiment, trained with the dataset having all the features. Optimization for LGBM was performed with the NSGA-II and found the following best combination of hyperparameter values, shown in Table of Appendix, fourth row, fourth column.

The F1-score of AdaB decreased from 0.973, the initial best model, to a value of 0.846. A similar phenomenon has happened with GBC. Probably the small number of features that have been selected for this model, related to the components of sensor 2, has caused a lack of useful information to reach higher performance.

XGB presented some good results that are not too distant from those obtained by the best model trained with the dataset having all the features. In this model were selected 28 features in which appeared at least 2 times the features related to each sensor. As a result, these results may be useful later to determine the best statistical index.

For each model, the selected features are shown in Table 4. The best model for AdaB produces the following features (19 in total); It is noteworthy that features related to sensor 2 no longer appear. In addition, the absence of features related to the median and the abundance of features related to the standard deviation can also be highlighted. This factor shows that some statistical indices are more useful than others. In increasing order of importance, statistical indices are median, minimum value, maximum value, standard deviation.

The GBC model selected 18 features. Similar to AdaB, there is no feature related to sensor 2, and no features corresponding to the median. A further consideration concerns sensor 4 because its features appear many times less than those of other sensors.

The XGB model selects a very large number of features equal to 28. It is noteworthy that features of all sensors appear, at least for 2 statistical indices.

Finally, for LGBM, that reaches the best performance, the features selected are 34.

In summary, the best F1-score result is 0.973 achieved with AdaB and GBC, while with feature selection techniques we reached the best F1-score of 0.919 with LGBM.

6.3 Discussion and Comparisons

Analyzing the results obtained in the two experiments, the optimization strategies, both by TPE and NSGA-II, produced very robust models. Despite the unbalanced dataset, the models were able to correctly classify samples belonging to all 12 classes. By maximizing the F1-score rather than accuracy, it was possible to guarantee the robustness of models in unbalanced cases. Furthermore, the

Table 2. Models Performance with four optimizations

Model	Optim	Hyperp	F1	Acc	Prec	Sens	Spec	G-Mean	ROC-AUC
AdaB	TPE	Few	0.961	0.957	0.962	0.962	0.996	0.979	0.998
AdaB	NSGA-II	Few	0.943	0.944	0.938	0.954	0.995	0.974	0.998
AdaB	TPE	All	0.973	0.969	0.975	0.973	0.997	0.985	0.996
AdaB	NSGA-II	All	0.967	0.963	0.968	0.968	0.997	0.982	0.997
GBC	TPE	Few	0.901	0.957	0.924	0.918	0.996	0.956	0.997
GBC	NSGA-II	Few	0.906	0.944	0.928	0.910	0.995	0.951	0.998
GBC	TPE	All	0.944	0.963	0.953	0.945	0.997	0.971	0.997
GBC	NSGA-II	All	0.973	0.969	0.974	0.973	0.997	0.985	0.996
XGB	TPE	Few	0.884	0.926	0.914	0.897	0.993	0.943	0.996
XGB	NSGA-II	Few	0.900	0.932	0.910	0.900	0.994	0.946	0.996
XGB	TPE	All	0.907	0.944	0.917	0.907	0.995	0.950	0.997
XGB	NSGA-II	All	0.901	0.938	0.923	0.901	0.994	0.947	0.996
LGBM	TPE	Few	0.854	0.920	0.883	0.849	0.992	0.918	0.997
LGBM	NSGA-II	Few	0.876	0.932	0.896	0.876	0.994	0.933	0.996
LGBM	TPE	All	0.937	0.951	0.949	0.938	0.995	0.966	0.998
LGBM	NSGA-II	All	0.939	0.957	0.947	0.940	0.996	0.968	0.996

Table 3. Performance of the Best Optimized Models trained with Feature Selection

Model	Optim	Hyperp	F1	Acc	Prec	Sens	Spec	G-Mean	ROC-AUC
AdaB	TPE	All	0.846	0.877	0.871	0.835	0.988	0.908	0.989
GBC	NSGA-II	All	0.877	0.883	0.876	0.884	0.989	0.935	0.991
XGB	TPE	All	0.808	0.870	0.901	0.780	0.988	0.878	0.989
LGBM	NSGA-II	All	0.919	0.914	0.937	0.911	0.992	0.951	0.992

Table 4. Selected features from each model

	Median		Min							Max							Std						
AdaB							5	6	7	1	3				7	1		3	4		6	7	
GBC					3	4		6	7				5	6	7	1		3		5	6	7	
XGB			1	2	3	4	5		7	1			5	6	7		2	3	4	5	6	7	
LGBM	2	5	1		3	4	5	6	7	1	3	4	5	6	7	1		3	4	5	6	7	

Table 5. Comparison of performance metrics in SHM damage classification

Author	Model	F1	Acc
Athanasiou et al. [3]	Boosted trees	–	42.9%
	Ensemble bagged trees	-	89.3%
	Subspace discriminant	–	78.5%
	Subspace KNN	–	80.1%
	RUSBoosted trees	–	71.4%
Bhatta et al. [6]	RF	0.736	74.62%
	SVM	0.666	64.17%
	DT	0.625	68.65%
	ANN	0.697	62.68%
	KNN	0.615	70.14%
Eltouny et al. [11]	Bayesian-optimized KDME	0.938	92%
Kostinakis et al. [18]	SVM - Gaussian Kernel	0.886	88.6%
	XGB	0.874	87.5%
	AdaB	0.765	76%
	LGBM	0.876	87.7%
The Proposed Work	AdaB (1st exp)	0.973	96.9%
	GBC (1st exp)	0.973	96.9%
	XGB (1st exp)	0.907	94.4%
	LGBM (1st exp)	0.939	95.7%
	AdaB (2nd exp)	0.846	87.7%
	GBC (2nd exp)	0.877	88.3%
	XGB (2nd exp)	0.808	80.8%
	LGBM (2nd exp)	0.919	91.4%

use of multiple evaluation metrics such as accuracy, sensitivity, precision, recall and ROC AUC is crucial for obtaining a complete and accurate understanding of the model’s performance. Each metric provides a different and complementary perspective on the model’s strengths and weaknesses. Considering the results obtained from both experiments, it can be noted that G-Mean is in most cases more than 0.93. Sensitivity reached values over 0.85. Similar results are evident from specificity and ROC AUC. Comparing the two experiments (Table 2 and Table 3), feature selection highlights how very similar results can be obtained by reducing the number of features. However, maintaining a feature space of 35–55% of the initial features, the F1-score decreased by 10%. Only LGBM, selecting about 65% of features, showed a decrease in F1-score of about 2%. The

monoaxial sensors affected the overall performance, so it is necessary to find a trade-off between the number of sensors used to acquire the data and the performance of the model. Mention should be made of the efficiency of sensor 3 with triaxial components, as was selected more frequently than sensors 1 and 2. The most incisive statistical index was the standard deviation due to the large number of features selected. While the worst was the median. As for the values of minimum and maximum, there were comparable results. The sensor that is always selected is the number 7. A crucial characteristic is the position of the sensor: sensors 6 and 7, positioned lower, had a greater impact, while sensors 1 and 2, positioned higher, were less effective. The obtained results suggest it could be useful to consider positioning sensors closer to the ground.

Table 5 shows a comparison of the results from different studies in the literature. The performance of F1-scores and accuracy varies between the types of models used in the detection of structural damage. It is noteworthy that the models from the proposed work and Eltouny et al. [11] obtained the best performance. More specifically, the Bayesian-Optimized KDME achieved an F1-score of 0.938, which is very close to the results of this work. On the other hand, Athanasiou et al. [3] showed very mixed results, with accuracy scores varying significantly from 89.3% for Ensemble bagged trees to 42.9% for Boosted trees. Bhatta et al. [6] models showed moderate performance, with RF achieving the best F1-score of 0.736 and an accuracy of 74.62%. Kostinakis et al. [18] models, such as SVM with Gaussian kernel and LGBM, showed good performance, with average accuracy of 88.6% and 87.7% respectively, and F1-scores of 0.886 and 0.876. Finally, this work demonstrates the effectiveness of the proposed approach, emphasizing the importance of feature selection and hyperparameter optimization in improving the performance of reinforced concrete structure damage classification models. The multidisciplinary approach adopted in this work, which integrates advanced machine learning, statistical optimization, and optimal sensory configurations, represents a significant step towards more effective and accurate solutions in the field of structural health monitoring.

7 Conclusions

In this study, we introduced a novel approach to feature selection voting strategies and a combination of hyperparameter optimization techniques to improve the performance of boosting classification for predicting damage in reinforced concrete frames. We proposed a new experimental pipeline to specify which types of sensors can be used and which ML techniques apply in damage assessment of concrete structures.

Through extensive experimentation and analysis, we demonstrated the effectiveness of the proposed methodology in accurately classifying different degrees of damage in concrete structures. Our results indicate that boosting algorithms offer significant advantages in terms of classification accuracy and efficiency.

Our investigation into optimization strategies has highlighted the importance of hyperparameter fine-tuning to achieve optimal performance. We used the TPE and NSGA-II methods because they represent a good compromise between computational cost and performance. A probabilistic approach is the best solution for hyperparameter search over a wide range of values for very complex models with long training times. Through careful optimization, we were able to further enhance the accuracy and reliability of the classification models, thereby improving their practical applicability in real-world scenarios.

The feature selection highlights how very similar results can be obtained by reducing the number of features (see comparison between Table 3 and Table 4). However, on average the F1-score decreases with a feature selection that eliminates half of the initial features. To maintain a satisfactory performance, it is necessary to find a trade-off between the number of sensors with which to acquire the data and the performance of the model.

Furthermore, the feature selection phase was not primarily aimed at enhancing the performance metrics of the model, but rather at elucidating which information is most significant for accurate classification. On one hand, improved computational efficiency facilitates the real-world implementation and utilization of these techniques. On the other hand, it sheds light on the sensors that are truly significant for classifying the structural state, reducing the number of sensors to be employed in future detection.

This is the first step in a study that could be complemented by validating the experimental setup in a real scenario. Overall, our findings contribute to advancing the field of structural health monitoring and maintenance by providing efficient and reliable methods for assessing damage in reinforced concrete structures. Future research directions may include exploring additional machine learning techniques and integrating advanced sensor technologies to further improve the performance of damage classification models.

A Appendix: Hyperparameter Values for Each Best Model

Classif	Hyperp	Value (1st Exp)	Value (2nd Exp)
AdaB	criterion	'gini'	
	max_depth	10	
	min_samples_split	19	
	min_samples_leaf	4	
	minWeightFractLeaf	0.000658323935614794	
	max_features	'sqrt'	
	max_leaf_nodes	25	
	min_imp_dec	0.0010919418200059734	
	ccp_alpha	1.5938666259963747e-05	
	n_estimators	120	
GBC	learning_rate	0.5851591914047651	
	learning_rate	0.01993085504537697	
	n_estimators	730	
	subsample	0.9500000000000001	
	min_samples_split	22	
	min_samples_leaf	18	
	minWeightFractLeaf	0.0018023265689062118	
	max_depth	11	
	min_imp_dec	0.001017736144910167	
	max_features	'sqrt'	
XGB	max_leaf_nodes	17	
	ccp_alpha	9.107190213356173e-07	
	n_estimators	360	
	max_depth	5	
	learning_rate	0.4754652930941692	
	tree_method	'approx'	
	gamma	0.0006339701705694259	
	min_child_weight	0.05301053790908121	
	max_delta_step	0.12177208860225514	
	subsample	0.7999999999999999	
LGBM	colsample_bytree	0.65	
	colsample_bylevel	0.7	
	colsample_bynode	0.6	
	reg_alpha	0.0015100503320114463	
	reg_lambda	35.498849933533876	
	num_leaves	115	160
	max_depth	9	3
	learning_rate	0.171355046221503	0.4704624041164966
	n_estimators	410	220
	subsample_for_bin	74	45
	min_split_gain	0.009511699517685778	0.002131321086577715
	min_child_weight	0.10002085367862634	0.9313446643260199
	min_child_samples	20	43
	subsample	0.8	0.9500000000000001
	subsample_freq	14	24
	colsample_bytree	0.8	0.8
	colsample_bynode	1.0	0.9
	max_delta_step	2.9780230033167596	0.38091374567573116
	reg_alpha	0.04567778071041187	0.32622555889971483
	reg_lambda	0.04567778071041187	62.74545963691049
	max_bin	250	300
	min_data_in_bin	16	5

References

1. Agarwal, D.A., Agrawal, A., Meruane, V., Sangwan, K.S.: Development of a machine learning based model for damage detection, localization and quantification to extend structure life. *Procedia CIRP* **98**, 199–204 (2021)
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: A next-generation hyperparameter optimization framework, Optuna (2019)
3. Athanasiou, A., Ebrahimkhanlou, A., Zaborac, J., Hrynyk, T., Salamone, S.: A machine learning approach based on multifractal features for crack assessment of reinforced concrete shells. *Comput.-Aided Civ. Infrastruct. Eng.* **35**(6), 565–578 (2020)
4. Bai, Y., Zha, B., Sezen, H., Yilmaz, A.: Engineering deep learning methods on automatic detection of damage in infrastructure due to extreme events. *Struct. Health Monit.* **22**, 338–352 (2023)
5. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: *Advances in Neural Information Processing Systems*, vol. 24 (2011)
6. Bhatta, S., Dang, J.: Seismic damage prediction of RC buildings using machine learning. *Earthquake Eng. Struct. Dyn.* **52**(11), 3504–3527 (2023)
7. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, New York, NY, USA, pp. 785–794. ACM (2016)
8. Chen, X., Jeong, J.C.: Enhanced recursive feature elimination. In: *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pp. 429–435 (2007)
9. De Buck, V., André Muñoz López, C., Nimmegeers, P., Hashem, I., Van Impe, J.: Multi-objective optimisation of chemical processes via improved genetic algorithms: a novel trade-off and termination criterion. In: Kiss, A.A., Zondervan, E., Lakerveld, R., Özkan, L. (eds.) *29th European Symposium on Computer Aided Process Engineering*, volume 46 of *Computer Aided Chemical Engineering*, pp. 613–618. Elsevier (2019)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
11. Eltouny, K., Liang, X.: Bayesian-optimized unsupervised learning approach for structural damage detection. *Comput.-Aided Civ. Infrastruct. Eng.* **36**, 05 (2021)
12. Figueiredo, E., Santos, A.: Chapter 1: Machine learning algorithms for damage detection. In: *Computational and Experimental Methods in Structures: Vibration-Based Techniques for Damage Detection and Localization in Engineering Structures*, pp. 1–39 (2018)
13. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) *EuroCOLT 1995*. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59119-2_166
14. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* 1189–1232 (2001)
15. Girden, E.R.: ANOVA: Repeated measures. Number 84. Sage (1992)
16. Ke, G., et al.: Lightgbm: a highly efficient gradient boosting decision tree. *Adv. Neural. Inf. Process. Syst.* **30**, 3146–3154 (2017)
17. Kendall, M.G.: A new measure of rank correlation. *Biometrika* **30**(1–2), 81–93 (1938)

18. Kostinakis, K., Morfidis, K., Demertzis, K., Iliadis, L.: Classification of buildings' potential for seismic damage by means of artificial intelligence techniques (2022)
19. Lazaridis, P.C., Kavvadias, I.E., Demertzis, K., Iliadis, L., Vasiliadis, L.K.: Structural damage prediction of a reinforced concrete frame under single and multiple seismic events using machine learning algorithms. *Appl. Sci.* (2022)
20. Lee, Y., Kim, H., Min, S., Yoon, H.: Structural damage detection using deep learning and FE model updating techniques. *Sci. Rep.* **13** (2023)
21. Liu, P.: Study on structural health monitoring practice using artificial intelligence. In: *Proceedings of the International Conference on Electrical, Automation and Computer Engineering (ICEACE), ICEACE '23*, pp. 804–809. IEEE (2023)
22. Oh, B.K., Jung, W.C., Park, H.S.: Artificial intelligence-based damage localization method for building structures using correlation of measured structural responses. *Eng. Appl. Artif. Intell.* **121** (2021)
23. Saarela, M., Jauhiainen, S.: Comparison of feature importance measures as explanations for classification models. *SN Appl. Sci.* **3**(2), 272 (2021)
24. Sharma, A., Mulik, N., Verma, R.P., Laxminarayananamma, K., Rao, A.L.N., Khan, A.K.: Prediction of structural health of civil engineering infrastructure using AI. In: *Proceedings of the 6th International Conference on Contemporary Computing and Informatics (IC3I), IC3II '23*, pp. 1595–1601. IEEE (2023)
25. Stone, M.: Cross-validated choice and assessment of statistical predictions. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **36**(2), 111–133 (1974)
26. Stone, M.: An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **39**(1), 44–47 (1977)
27. Watanabe, S.: Tree-structured parzen estimator: understanding its algorithm components and their roles for better empirical performance (2023)



Advancing 3D Mesh Analysis: A Graph Learning Approach for Intersecting 3D Geometry Classification

Stefan Andreas Böhm^{1,2} , Martin Neumayer², Bare Luka Zagar¹,
Fabian Riß², Christian Kortüm², and Alois Knoll¹

¹ Technical University of Munich, Boltzmannstraße 3, Garching 85748, Germany
{stefan-andreas.boehm,bare.luka.zagar,k}@tum.de

² Rosenheim Technical University of Applied Sciences,
Hochschulstraße 1, Rosenheim 83024, Germany

{martin.neumayer,fabian.riss,christian.kortuem}@th-rosenheim.de

Abstract. Driven by increasing customer demands, manufacturing processes now encompass increasingly intricate workflows. The industry uses computer-aided process planning to manage these complex manufacturing processes effectively. A crucial task here is to analyze product data and determine the required machining features, represented as 3D mesh geometries. However, a notable challenge arises, particularly with custom products, where the interpretation of the 3D mesh geometry varies significantly depending on the available machinery and expert preferences. This study introduces a configurable automated feature recognition framework based on expert knowledge. Experts can use a configurable synthetic data generator to encode their requirements within this framework via the training data. A machine-learning graph classification approach is used to recognize the 3D geometries of machining features in the generated data, based on to the user requirements. The system accomplishes this without requiring for data conversion into alternative formats, such as voxel or pixel representations, like other approaches are forced to.

Keywords: Intersecting 3D Meshes · Graph Classification · Graph Neural Networks

1 Introduction

Despite technological advances, most companies operate with static, centralized manufacturing processes. Due to inflexibility and lack of resource efficiency, many companies risk economic difficulties, especially during an economic crisis, as they cannot adapt to new circumstances quickly. Many small and medium-sized enterprises (SMEs) continue to use such static processes and are most affected during economic crises. Additionally, SMEs face the challenge of meeting the increasing demand for individualized products from customers with distinct and diverse preferences [13].

Therefore, intelligent systems are needed to enhance efficiency and enable the manufacturing of customer-specific products. These intelligent systems depend on the underlying infrastructure of cyber-physical systems (CPS), where components such as actuators and sensors can autonomously share data across a company's network. This forms the crucial framework for seamless communication and interaction between physical and digital elements. While CPS can autonomously carry out manufacturing tasks, a critical requirement for planning these tasks is the complete understanding of the given product [6]. Only then can the product's design be allocated to appropriate machines and tools. However, to do so, deep expert knowledge is required [2].

CAD data, used to allocate the product design to the appropriate equipment, is often represented by basic geometric features such as vertices and their edges, as shown in Fig. 1 [1]. Achieving an optimized manufacturing process depends on accurately recognizing these basic geometries within a CAD model as 3D representations of machining features. This classification is often called automated feature recognition (AFR) [11].

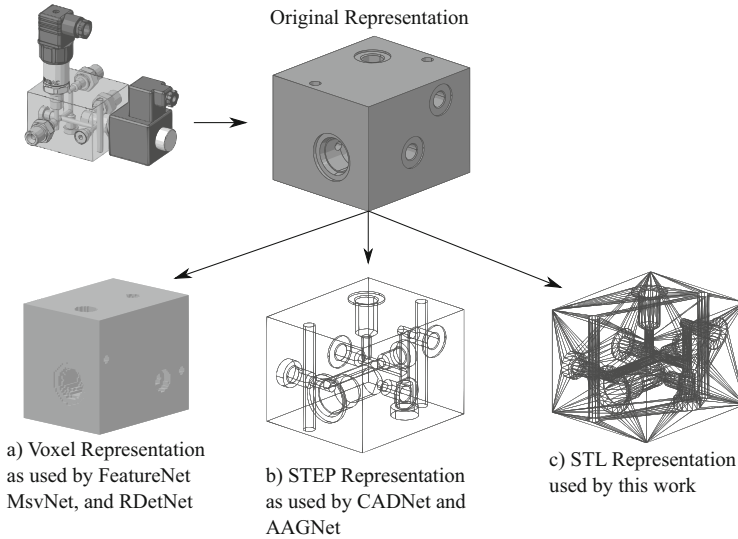


Fig. 1. Example of a hydraulic manifold in three different CAD representations. a) Voxel represents a 3D object as a cube-shaped grid. b) STEP (standard for the exchange of product data) is a format giving an exact description of each surface, its edges, and other essential details. c) STL (stereolithography) contains only a list of 3D vertices and their connections to other vertices. Inspired by [7, 19, 20, 27, 29].

Many current approaches for AFR focus on the CAD representations Voxel or STEP as shown in Fig. 1. In Voxel, the original CAD model is converted into a cube grid with a fixed resolution, similar to image processing. This allows for the analysis of CAD files with conventional methods such as 3D convolutional neural

networks. However, this conversion often loses essential information due to low resolution, as shown in Fig. 1a. Therefore, modern approaches increasingly focus on the widely used STEP format, which contains rich and accurate information about each surface of the CAD model, as shown in Fig. 1b. Another common CAD format, especially in additive manufacturing and virtual and augmented reality applications, is 3D meshes. Further, this work solely focuses on the 3D mesh format STL as shown in Fig. 1c, which only contains 3D vertice information and their connection to other vertices. Also, with the increasing geometric complexity of a 3D model, the amount of vertices drastically increases as well. Therefore, Colligan et al. in [7] suggest that the “performance could be boosted by tailoring the neural architecture on the mesh level for learning from mesh data or by directly learning the dihedral angle between the faces on the B-Rep level”.

This work follows the problem statement of Colligan et al.. It introduces a configurable AFR (CAFR) framework with a customized graph neural network (GNN) architecture for the machining feature recognition task in STL data. The framework also includes a configurable synthetic data generation strategy integrating user requirements into the training data generation process.

The remainder of this paper is organized as follows: Sect. 2 reviews recent related works. Section 3 provides a comprehensive overview of the methodology and experiments conducted in this study. Section 4 presents and discusses the experimental results. Finally, Sect. 5 concludes with a summary of the framework’s effectiveness and outlines future research.

2 Related Work

Various AFR approaches have been established over the past two decades, mainly within academia. These advancements have predominantly centered around rule- or hint-based, graph-based, volumetric decomposition, and machine learning approaches [22]. In recent years, attention has mainly shifted towards machine learning-based frameworks due to the limitations of deterministic or heuristic approaches, such as learning and generalizing new machining features and meeting computational requirements. With FeatureNet, Zhang et al. [29] introduced an innovative machine-learning AFR framework capable of accurately recognizing intersecting machining features. FeatureNet provides an extensive training dataset consisting of 24,000 CAD models. These models encompass 24 machining feature categories, each containing 1,000 CAD models with randomly placed features. FeatureNet transforms the CAD models into a voxel representation with a consistent resolution to address the problem of varying data sizes in STL CAD data. Finally, it employs a 3D convolutional neural network (CNN) as its supervised learning algorithm.

However, the research of Su et al. [24] demonstrated the potential of a multi-sectional view (MSV) representation for 3D object recognition. Combining multiple views of a 3D model through a neural network can achieve enhanced object recognition. Peizhi et al. [20] built upon this concept to introduce the MsvNet

framework. MsvNet collects and consolidates MSVs of a 3D model into a 2D convolutional neural network. This fusion of MSVs significantly improves classification accuracy and reduces the required training data size compared to FeatureNet.

Wu et al. [26] follow the principle and enhance the existing FeatureNet and MsvNet frameworks using a lightweight network technique inspired by the self-supervised learning method SimSiam [5]. This approach seeks to increase the efficiency of these networks, resulting in what Wu et al. term FeatureNetLite and MsvNet Lite.

FeatureNet, MsvNet, and their adaptations operate as two-stage procedures that first preprocess CAD models by segmenting them with unsupervised algorithms and then analyzing the segmented features. This approach results in significantly longer processing times and lower recognition accuracy. Consequently, Pei Shi et al. propose a single-stage AFR framework. This framework uses data augmentation to generate a comprehensive multi-machining feature dataset, which is then used to train a novel neural network architecture called RDetNet. Due to the single-stage approach, RDetNet is considerably faster than FeatureNet or MsvNet and demonstrates significantly better recognition accuracy [21].

Approaches like FeatureNet and MsvNet, along with their adaptations, face the challenge of converting CAD data into resolution-dependent representations, such as voxels or pixels, due to the requirements of their network architectures. Unfortunately, this conversion leads to losing crucial topological information due to low resolution. To address this limitation, Jia et al. [14] present an AFR framework where CAD models are transformed into a mesh structure. This innovative approach allows for the analysis of machining features using the mesh convolutional neural network [12] in combination with the fast region-based convolutional network [10] while preserving the valuable topological details. The input to the models consists of each mesh face's center, the vertices' angles at this center, and the indices of the adjacent mesh faces.

CAD models can also be interpreted as graph representations [9]. Here, a graph G is defined as $G = (V, E)$, where $V = \{V_1, \dots, V_n\}$ represents a collection of distinct nodes (further called vertices), and $E = \{\{V_k, V_w\}, \dots, \{V_i, V_j\}\}$ represents a collection of edges, indicating the connections between the distinct vertices [23]. The adjacency matrix, shown in Eq. 1, is a more compact representation of a graph, where A represents the graph $G = (V, E)$ as a matrix of size $|V| \times |V|$,

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1j} \\ A_{21} & A_{22} & \cdots & A_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{ij} \end{bmatrix} \quad (1)$$

where $A_{ij} = 1$ if there is a connection between vertex i and vertex j , and $A_{ij} = 0$ otherwise [23].

The approach of Colligan et al., called CADNet [4, 7], utilizes this graph representation principle. It considers individual faces of a STEP CAD model as graph nodes, with shared edges of two faces treated as graph edges. Additionally, while previous AFR frameworks rely on datasets containing CAD models with precisely one machining feature per model and employ data augmentation strategies to transform the training data into multi-feature data, CADNet introduces a synthetic data generator capable of producing STEP CAD models based on specifications from FeatureNet. CADNet uses these generated STEP CAD models to train the dynamic graph CNN from Verma et al. [25].

Similar to CADNet, Hongjin Wu et al. introduced another AFR framework based on the graph representation principle for STEP CAD models. However, unlike CADNet, Hongjin Wu et al. introduce a novel graph structure that combines UV grids of surfaces and curves to present geometric information, along with a Face Adjacency Graph (FAG) to encode topological information. This advanced graph structure is utilized to train a novel graph neural network architecture called AAGNet, aimed at semantic segmentation, instance segmentation, and bottom face identification in neutral STEP CAD models [27].

AFR frameworks based on GNN, such as AAGNet and CADNet, face inherent limitations, including restricted receptive fields and network depths due to the foundational structure of their underlying message-passing neural network architecture. To overcome these limitations, Zhang et al. introduced BrepMFR, which leverages recent advances in graph transformers to enhance the neural network’s feature extraction capabilities for more complex STEP CAD models [28].

3 Methodology

Following the basic principle of CADNet, this paper introduces a configurable automated feature recognition (CAFR) framework to address the problem of intersecting machining features in STL CAD data as shown in Fig. 1. However, unlike CADNet, this work focuses on recognizing the contained geometries not by surfaces but by the vertices and edges. For this purpose, a methodology is introduced to generate training data synthetically following user requirements, and this data is used to learn the task of AFR using a GNN. The CAFR framework is depicted in Fig. 2. In the following paragraphs, we describe this work’s threefold methodology.

3.1 Generating Synthetic Training Data with Intersecting Machining Feature

A substantial dataset of CAD models is required to train a GNN on the AFR problem. While FeatureNet provides such a dataset with 24,000 CAD models, each model in this dataset contains only one machining feature. Additionally, FeatureNet does not offer CAD model data augmentation functionalities such as rotation, translation, or scaling, which are necessary to create intersecting

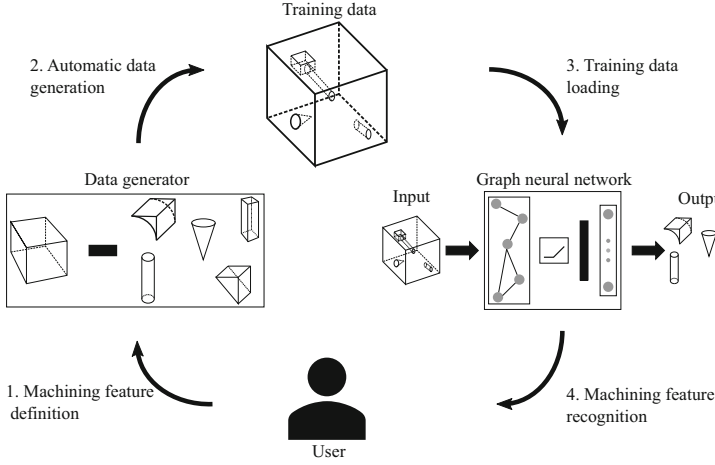


Fig. 2. Presentation of the CAFR framework with user experience integration.

machining features. CADNet’s data generation strategy follows the specifications of FeatureNet. However, it utilizes freely positionable sketches of machining features to create STEP CAD models with intersecting machining features, thereby addressing the limitations of the FeatureNet dataset. Nevertheless, the data generation strategy from CADNet exclusively produces STEP CAD models, for which state-of-the-art approaches like FeatureNet and MsvNet are not applicable.

Hence, we introduce a synthetic data generation strategy for STL CAD models as an integral component of the CAFR framework. This synthetic data generator relies on constructive solid geometry (CSG) modeling, often used for STEP and STL CAD models. An essential part of CSG is the composition operations: union $A \cup B$, intersection $A \cap B$, and difference $A \setminus B$ [1]. These composition operations are applied to 3D primitives and basic geometries such as cubes, cylinders, or pyramids to create new 3D objects. Upon closer examination, the FeatureNet dataset can be partitioned into ten distinct geometric primitives, shown in Fig. 3.

By implementing configurable CSG decomposition techniques, as exemplified by the synthetic data generation strategy illustrated in Fig. 4, this innovative approach empowers the generation of datasets in alignment with the state-of-the-art requirements described by FeatureNet and MsvNet. This data generation strategy also allows the generation of STL CAD Data with intersecting machining features as depicted in Fig. 1.

3.2 Validating the Synthetic Training Data Generation Strategy

Given that the STL CAD data from the process depicted in Fig. 4 form the foundation of this study, the generated data must first be validated. Here, validating the generated STL CAD models necessitates a

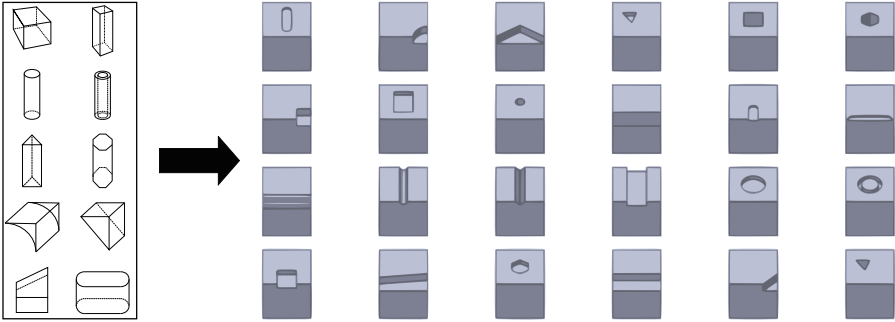


Fig. 3. Example of machining features from the FeatureNet dataset: Each of the 24 cubes displayed contains a specific machining feature with specifications derived from real manufacturing processes. The entire FeatureNet dataset can be partitioned into ten distinct geometric primitives. Inspired by [29].

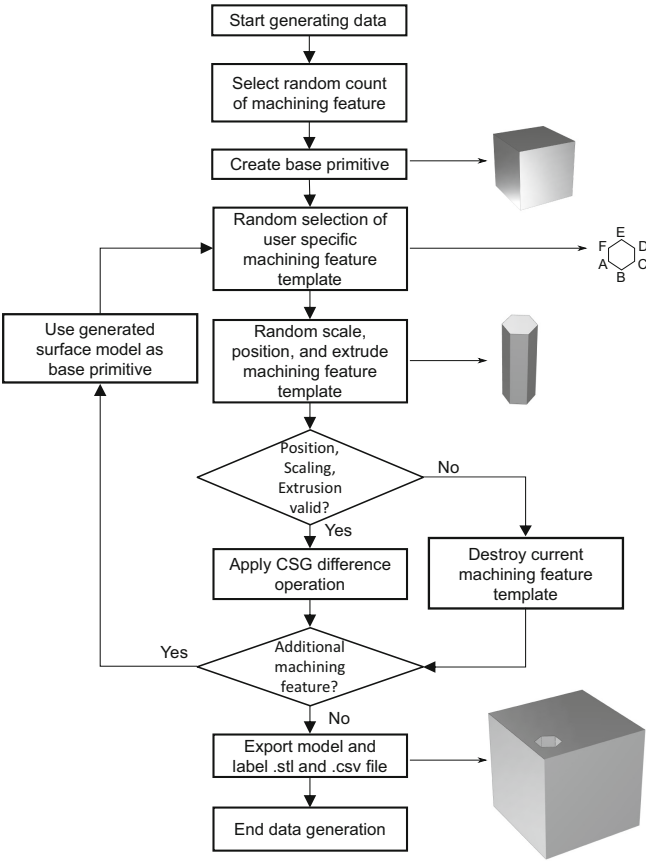


Fig. 4. Procedure of the CAFR data generation strategy of synthetic data. The strategy builds on the python package PyMadCad [3].

comprehensive assessment to confirm the complete closure of their surfaces and, therefore, the absence of data corruption.

To guarantee the accurate and enclosed surface generation of STL CAD models, the initial step of this study involves their conversion into a voxel representation. Here, the voxel conversion method, BINVOX, introduced by Patrick Min et al. [15, 17], is used. The successful application of BINVOX in this context verifies the correctness of the generated CAD models. It also establishes foundational compatibility with other AFR frameworks like FeatureNet.

To ensure that the synthetic data generator follows the machining feature definition established by FeatureNet, state-of-the-art AFR frameworks for single feature recognition, including FeatureNet, MsvNet, FeatureNetLite, and MsvNet Lite, were trained on the unrotated 21.600 FeatureNet CAD models and then tested on 2400 unrotated single machining feature dataset created by the CAFR frameworks and the original FeatureNet dataset. This procedure targets the exposure of any inconsistencies between the data strategy of this work and the FeatureNet dataset. This work uses the provided source code of each framework by Wu et al. [26] to evaluate the aforementioned state-of-the-art AFR frameworks.

3.3 Evaluating Graph Neural Network Applicability for Intersecting Machining Feature Recognition

Suppose STL CAD models contain multiple machining features; in that case, intersections may occur, as depicted in Fig. 1. These intersections often result in the deletion of original machining feature vertices, leading to the loss of crucial topological information. This loss significantly impacts the recognition accuracy of AFR frameworks.

The effectiveness of GNNs for STL CAD models with intersecting machining features must be evaluated to address this challenge. Given that AFR involving intersecting machining features necessitates multi-label classification of the entire CAD model, we have implemented the k-dimensional Graph Neural Network (k-GNN) as detailed in [16], using the PyTorch Geometric framework [8], for graph classification. The k-GNN specializes in classifying an entire graph by labeling each node individually and then aggregating these labels through a global mean pooling layer to represent the entire graph. This network excels in categorizing labeled graphs and primarily processes sets of graph nodes as its input. At each layer, the k-GNN computes a unique feature vector for each node set, enhancing its ability to detect intricate patterns. To ensure scalability and prevent overfitting in large datasets, the k-GNN employs a strategy that strategically excludes the global neighborhood of the current node set. This approach is supported by a hierarchical structure, where features learned by a (k-1)-dimensional GNN augment the network's overall effectiveness. Additionally, the k-GNN utilizes sampling strategies to manage larger datasets effectively, facilitating efficient scaling while maintaining robust performance.

Hyperparameter optimization is imperative to adapt the described k-GNN effectively to the challenge of intersecting machining features. To accomplish this objective, we employ Optuna’s hyperparameter optimization framework. Optuna uses the hyperparameter intervals from Table 1 to optimize a network architecture to the given dataset. In this work, multiple training cycles are conducted. Optuna optimizes the given hyperparameters after every training cycle to maximize the F1 score on the validation dataset. For the hyperparameter optimization of the k-GNN model, we utilize the parameters and value ranges depicted in Table 1.

Table 1. Hyperparameter table of the k-GNN with suggested value intervals by Moris et al. [16].

Hyperparameter	Suggested Values
Number of graph convolutional layers	2; 3; 4; 5; 6; 7
Number of hidden channels	32; 64; 128; 256; 512
Batch size	16; 32; 64; 128; 256
Learning rate	0.01; 0.001; 0.0001
Dropout probability	0.1; 0.2; 0.3; 0.4; 0.5

Regarding the necessary training data, this work follows the guidelines of FeatureNet and MsvNet for the problem of intersecting machining feature recognition, where 144,000 CAD models with up to 8 intersecting machining features per model are created using the data generation strategy of the CAFR framework. Additionally, eight test datasets containing 177 STL CAD models each are created with the CAFR data generator. Each dataset comprises CAD models with a fixed number of machining features, varying from 1 to 8. The trained k-GNN is compared with MsvNet, which shows high generalizability on different datasets as depicted in Table 2. Also, due to the MSV strategy, MsvNet can utilize 147,456 training images combined with preprocessing segmentation. This results in a high recognition performance for intersecting machining features and,

Table 2. Accuracy comparison of state-of-the-art AFR frameworks on test models from FeatureNet and the CAFR framework. The validation accuracy is based on 100 voxel models per machining feature class from the unrotated FeatureNet dataset and the CAFR framework.

Dataset	Automatic Feature Recognition Frameworks			
	voxel based		pixel based	
	FeatureNet	FeatureNetLite	MsvNet	MsvNetLite
FeatureNet \uparrow	95.00%	89.92%	96.00%	93.62%
CAFR \uparrow	89.38%	82.50%	96.12%	88.12%
Difference \downarrow	5.62%	7.42%	0.12%	5.5%

therefore, qualifies MsvNet as a well-suited benchmark. Also, the CAFR framework is tested on a real hydraulic manifold to test its applicability to real-world applications. For a fair comparison, this work omits approaches like SsdNet, which utilize data augmentation techniques to enhance the training data size to an excessive two million images.

4 Results and Discussion

For the following experiments, the source code of the CAFR framework with the detailed GNN architectures, which was used to generate the results, can be found at the [Github repository: CAFR framweork](#). The source code for the state-of-the-art approaches can be found in the respective references.

For compatibility reasons, all experiments were conducted on a system equipped with an Intel Core i7-10850H processor, 32 GB of RAM, and an Nvidia Quadro T2000 with 16GB of memory, which still supports CUDA 10.2. This setup was necessary to accommodate the state-of-the-art AFR framework implementations and the outdated TensorFlow versions they required.

4.1 Validation of the Synthetic Generated Training Data

In order to verify the functionality of the synthetic data generator within the CAFR Framework, the approaches detailed in Sect. 3.1 and Sect. 3.2 were employed. Here, the entire FeatureNet dataset was converted into a voxel format with a resolution of $64 \times 64 \times 64$ and then used for training. Following the guidelines of Wu et al. [26], the training dataset was divided, using 900 voxel models for training purposes and 100 models for validation in each machining feature class. Figure 5 depicts the training and validation results, demonstrating that all AFR frameworks, except for FeatureNetLite, achieve a recognition accuracy exceeding 90%. However, it should also be noted that the frameworks FeatureNetLite exhibit slight indications of overfitting despite utilizing the complete unrotated FeatureNet dataset comprising 21,600 training models.

To validate the accuracy of the synthetic data generator, a test dataset comprising 100 STL CAD models for each machining feature class was created using the CAFR framework, comprising 2400 CAD models in total. These models were transformed into a $64 \times 64 \times 64$ voxel format using the BINVOX software, successfully achieving voxel conversion. This BINVOX conversion confirms the correctness of the model surfaces. Subsequently, we tested state-of-the-art AFR frameworks, trained on the unrotated FeatureNet dataset, on this test data and compared their accuracy with the validation accuracy on the FeatureNet data. The outcomes of these evaluations are detailed in Table 2.

The table shows that MsvNet achieves the highest recognition accuracy, reaching 96%. Overall, The results relate to Wu et al.'s findings in [26]. Additionally, the table indicates that the advanced frameworks demonstrate similar recognition accuracy on the data generated by the CAFR framework. However, the approaches FeatureNet, FeatureNetLite, and MsvNetLite show differences in

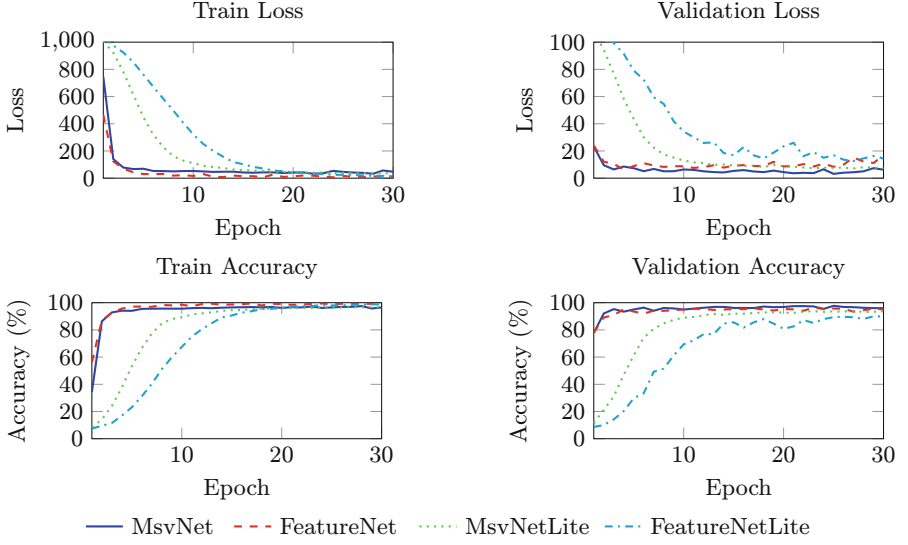


Fig. 5. Training Loss, Validation Loss, Training Accuracy, and Validation Accuracy of the AFR frameworks MsvNet, FeatureNet, MsvNetLite, and FeatureNetLite are reported. Each framework was trained for 30 epochs on 900 non-rotated CAD models from each of the 24 machining feature classes derived from the original FeatureNet dataset. Additionally, 100 CAD models per class were used to assess the accuracy of the validation. The training of these frameworks utilized the implementation by Wu et al. [26], specifically tailored to address the problem of recognizing single machining features.

accuracy of around 5–7% compared to the CAFR dataset, which can be explained by some mislabelled machining feature classes in categories eight and twelve. Despite the minor variations between the two test datasets, the consistently high recognition accuracy levels, particularly in the case of MsvNet, validate the correct synthetic data generation process within the CAFR framework.

4.2 Results of Graph Neural Networks for Intersecting Feature Recognition

A novel method for generating and validating training data for the AFR problem has been introduced up to this point. The next and final stage of this work involves training the k-GNN on the problem of AFR with intersecting machining features. For this purpose, we use a training dataset generated through the CAFR framework and validated via BINVOX conversion. This dataset comprises 144,000 training and validation models, as suggested by Peizhi et al. [20]. The training and validation data ratio is 80% training and 20% validation data. It includes STL CAD models containing up to eight intersecting machining features. A hyperparameter optimization with 100 training cycles was conducted using the parameters listed in Table 2 to assess the impact of missing vertices

due to these intersections. We conduct a cross-validation mechanism after every training cycle, in which the complete dataset is randomly shuffled and then split again in training and validation datasets with the described ratio.

The best-performing network architecture of the hyperparameter optimization concluded a network architecture with seven k-GNN layers where each has a hidden layer size of 512. The batch size is 32, the learning rate is 0.001, and the dropout probability is 0.2. As the final classifier of the architecture, a global mean pooling layer with a following linear layer is used. For the multi-label classification task, the BCEWithLogitsLoss function is applied. The training and validation results of this architecture are shown in Fig. 6.

With 80% validation accuracy for the highly complex problem of intersecting machining features recognition, the k-GNN shows promising potential for the task at hand. To further analyze the applicability of the k-GNN, the following comparative analysis in Table 3 showcases its strengths and limitations compared to the state-of-the-art AFR frameworks MsvNet for intersecting feature recognition. Here, MsvNet is chosen due to its high generalizability on different datasets, as shown in Table 2. The k-GNN shows a significantly higher recognition accuracy, especially with few machining features. However, it is essential to highlight that the k-GNN, in contrast to MsvNet, can directly process STL CAD data without converting it into voxel or pixel representation with a low resolution. Additionally, a particularly notable point is the runtime of the k-GNN compared to FeatureNet and MsvNet; the k-GNN is, on average, 70 times faster. This can be attributed to the fact that MsvNet must process 12 times a high pixel resolution of 256×256 for each CAD model due to the applied Msv

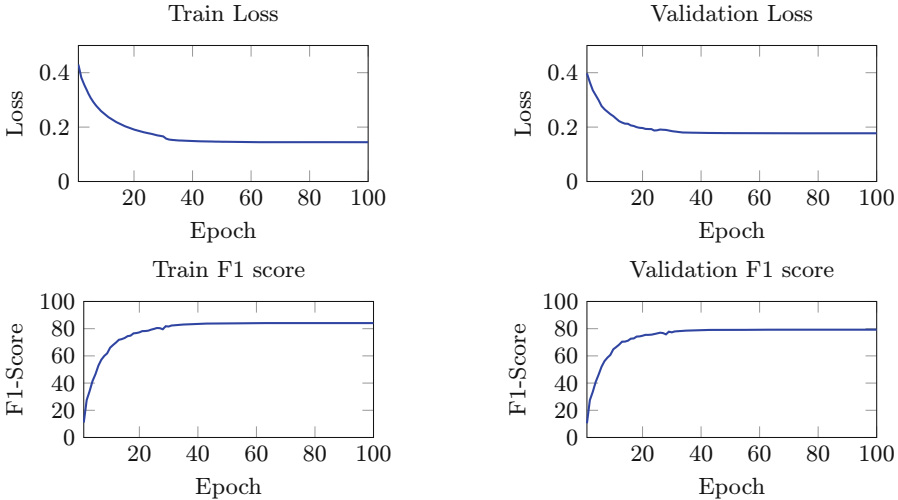


Fig. 6. Training Loss, Validation Loss, Training Accuracy, and Validation Accuracy of the k-GNN on a dataset with intersecting machining feature

Table 3. Representation of the micro F1-score and runtime performance on the test data for MsvNet and the k-GNN (ours). The best results are shown in bold.

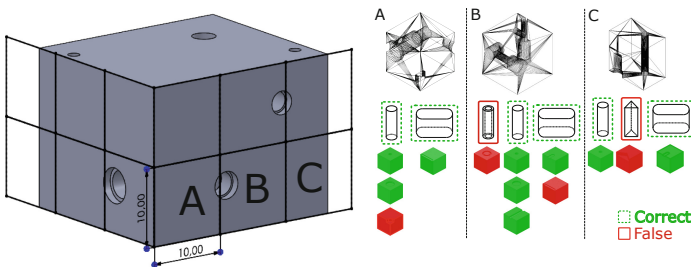
MFR	Machinig Feature in each test group						
Frameworks	2	3	4	5	6	7	8
MsvNet							
F1 Score \uparrow	79.34%	74.74%	65.56%	61.45%	54.11%	55.26%	51.00%
Runtime \downarrow	121.72	160.71 s	184.87 s	249.04 s	270.97 s	370.26 s	324.36 s
k-GNN							
F1 Score \uparrow	93.36%	90.03%	84.09%	79.11%	75.50%	74.93%	69.00%
Runtime \downarrow	3.22 s	3.17 s	3.22 s	3.31 s	3.35 s	3.43 s	3.76 s

technique. In contrast, the k-GNN, with an average of 3,033 vertices, requires significantly less computational effort.

4.3 Generalizability and Scalability of Graph Neural Networks to Real World Application

This work addresses the AFR problem based on the theoretical foundation established by FeatureNet. In its nature, however, AFR is a practical problem. Therefore, a prototypical approach to analyzing manufacturing parts from various industry branches using the CAFR framework is presented. For example, the hydraulic manifold [18] introduced in Fig. 1 was used for the real-world applicability test.

Since the k-GNN of the CAFR framework is trained exclusively on cube-shaped STL CAD data, it is necessary to adjust real components accordingly to maintain consistent performance. For this purpose, a simple strategy based on the boolean difference operation is used. A grid with individual 10 mm cubes is created, as shown in Fig. 7. The individual cubes are iterative differentiated from the original component. This simple strategy allows components with various

**Fig. 7.** A prototypical approach to analyze various components from different industrial sectors using the CAFR framework.

shapes and structures to be converted into the necessary basic geometry for the CAFR framework.

In Fig. 7, the segmented cubes A, B, and C are presented as test results. The CAFR framework can generally recognize the basic geometries. However, two basic geometries were incorrectly identified in cubes B and C. In cube B, a stepped blind hole was not included in training data due to the FeatureNet specifications and, therefore, was wrongly associated with an o-ring feature. In cube C, a triangular blind step was incorrectly identified. This can be explained by cube C being an edge piece of the component, only 3 mm wide, with slight intersections in the middle that could be interpreted as the tip of a triangular blind step. Additionally, a circular blind step and a horizontal circular end blind slot were classified as false positives, likely due to the significant intersections at the blind holes, similar to these false positive features. Despite the significant intersections, the remaining features were correctly identified, indicating potential real-world applications.

5 Conclusion

In the context of this study, a comparative analysis between the GNNs and the state-of-the-art framework MsvNet for the problem of intersecting machining features in STL CAD data was conducted. The objective was to determine whether GNNs, particularly the unique k-GNN architecture, could effectively classify machining feature patterns in STL CAD models. This study also involved the development of a customized k-GNN architecture and a configurable data generation strategy for STL CAD data in regard to the AFR problem, as exemplified in Fig. 8.

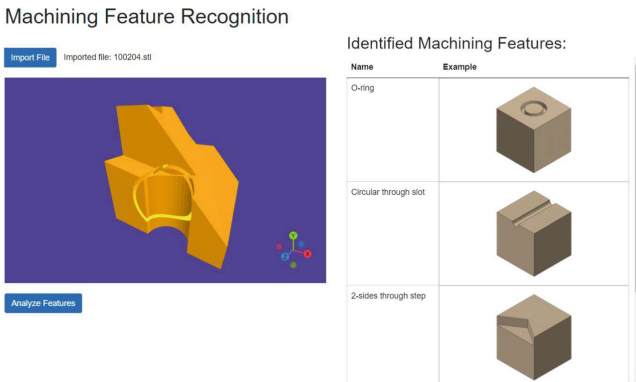


Fig. 8. GNN based AFR for intersecting machining feature.

One of the particular strengths of state-of-the-art approaches like MsvNet is their ability to correctly classify machining features despite significant intersec-

tion with other machining features. The comparative study in Sect. 4.2 demonstrates that the k-GNN of this work’s CAFR framework outperforms MsvNet in all scenarios without the need to convert the CAD data into a simpler 3D format. It is particularly noteworthy that the k-GNN demonstrates a remarkable advantage over the state-of-the-art framework in terms of runtime performance, thereby highlighting its efficiency.

In addition to the AFR problem defined by FeatureNet, Sect. 4.3 demonstrates the applicability of the CAFR framework to real-world data using an industrial hydraulic manifold. The hydraulic manifold is divided into suitable cube structures using a simple boolean difference strategy to match the specifications of the training data. The k-GNN of the CAFR framework can recognize all known basic geometries despite significant machining feature intersections. Only when machining features deviate significantly from the original training dataset occasional false positives occur. Overall, the simple boolean difference strategy for component segmentation and the fast runtime of the k-GNN, demonstrate the potential of the CAFR framework for real-world applications in various sizes and shapes.

However, it is important to note that further development is required for an accurate and stable industrial application. Specifically, the integration of component segmentation for applying real-world components in the CAFR framework as an automated preprocessing step is a crucial area for future development. The k-GNN architecture, built for graph classification, can only recognize machining features. Future work should also focus on node classification, allowing for machining feature segmentation alongside feature recognition. Additionally, the focus should shift from the machining features defined by FeatureNet to those that represent more realistic industrial applications. This requires an application study across various independent industrial fields. Also, the CAFR framework should not only rely on synthetic data generation in the future but also incorporate data augmentation strategies such as graph rotation, scaling, and node translation. This could enhance the robustness of the CAFR framework concerning different vertex topologies, which can vary between different CAD software.

Acknowledgements. This work was funded by the Bavarian State Ministry of Economic Affairs, Regional Development, and Energy as part of the Bavarian Joint Funding Program (BayVFP) - Digitization funding line - Information and Communication Technology funding area. The BayWISS Joint Academic Partnership Digitalisation also supported Stefan Andreas Böhm.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process. During the preparation of this work, the authors used ChatGPT and grammarly.com to ensure the correct spelling in this work. After using ChatGPT and grammarly.com, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

1. Bi, Z., Wang, X.: Computer Aided Design and Manufacturing. Wiley (2020)
2. Bozarth, C.C., Warsing, D.P., Flynn, B.B., Flynn, E.J.: The impact of supply chain complexity on manufacturing plant performance. *J. Oper. Manag.* **27**(1), 78–93 (2009)
3. jimy byerley: pymadcad (2023). <https://pymadcad.readthedocs.io/en/latest/>
4. Cao, W., Robinson, T., Hua, Y., Boussuge, F., Colligan, A.R., Pan, W.: Graph representation of 3D cad models for machining feature recognition with deep learning. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. vol. 84003, p. V11AT11A003. American Society of Mechanical Engineers (2020)
5. Chen, X., He, K.: Exploring simple SIAMESE representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758 (2021)
6. Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., Fraise, P.: Collaborative manufacturing with physical human-robot interaction. *Robot. Comput.-Integrat. Manuf.* **40**, 1–13 (2016)
7. Colligan, A.R., Robinson, T.T., Nolan, D.C., Hua, Y., Cao, W.: Hierarchical cad-net: learning from b-reps for machining feature recognition. *Comput. Aided Des.* **147**, 103226 (2022)
8. Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. arXiv preprint [arXiv:1903.02428](https://arxiv.org/abs/1903.02428) (2019)
9. Flynn, P.J., Jain, A.K.: Cad-based computer vision: from cad models to relational graphs. In: Conference Proceedings, IEEE International Conference on Systems, Man and Cybernetics, pp. 162–167. IEEE (1989)
10. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
11. Han, J., Pratt, M., Regli, W.C.: Manufacturing feature recognition from solid models: a status report. *IEEE Trans. Robot. Autom.* **16**(6), 782–796 (2000)
12. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D.: MeshCNN: a network with an edge. *ACM Trans. Graph. (ToG)* **38**(4), 1–12 (2019)
13. Hillary, R.: Small and Medium-Sized Enterprises and the Environment: Business Imperatives. Greener management international, Taylor & Francis Group (2000). <https://books.google.de/books?id=Dj6RAQAACAAJ>
14. Jia, J.L., Zhang, S.W., Cao, Y.R., Qi, X.L., WeZhu: Machining feature recognition method based on improved mesh neural network. *Iranian J. Sci. Technol. Trans. Mech. Eng.* 1–14 (2023)
15. Min, P.: binvox. <http://www.patrickmin.com/binvox> or <https://www.google.com/search?q=binvox> (2004 - 2019). Accessed 19 Sept 2003
16. Morris, C., et al.: Weisfeiler and leman go neural: Higher-order graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4602–4609 (2019)
17. Nooruddin, F.S., Turk, G.: Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Visual Comput. Graphics* **9**(2), 191–205 (2003)
18. Rolinck, N., Schmitt, M., Schneck, M., Schlick, G., Schilp, J.: Development workflow for manifolds and fluid components based on laser powder bed fusion. *Appl. Sci.* **11**(16), 7335 (2021)
19. Shi, P., Qi, Q., Qin, Y., Scott, P.J., Jiang, X.: Intersecting machining feature localization and recognition via single shot multibox detector. *IEEE Trans. Industr. Inf.* **17**(5), 3292–3302 (2020)

20. Shi, P., Qi, Q., Qin, Y., Scott, P.J., Jiang, X.: A novel learning-based feature recognition method using multiple sectional view representation. *J. Intell. Manuf.* **31**(5), 1291–1309 (2020). <https://doi.org/10.1007/s10845-020-01533-w>
21. Shi, P., Qi, Q., Qin, Y., Scott, P.J., Jiang, X.: Highly interacting machining feature recognition via small sample learning. *Robot. Comput.-Integr. Manuf.* **73**, 102260 (2022)
22. Shi, Y., Zhang, Y., Xia, K., Harik, R.: A critical review of feature recognition techniques. *Comput.-Aided Des. Appl.* **17**(5), 861–899 (2020)
23. Stamile, C., Marzullo, A., Deusebio, E.: *Graph Machine Learning: Take Graph Data to the Next Level by Applying Machine Learning Techniques and Algorithms*. Packt Publishing Ltd. (2021)
24. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 945–953 (2015)
25. Verma, N., Boyer, E., Verbeek, J.: FeastNet: feature-steered graph convolutions for 3D shape analysis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2598–2606 (2018)
26. Wu, H., Lei, R., Huang, P., Peng, Y.: A semi-supervised learning framework for machining feature recognition on small labeled sample. *Appl. Sci.* **13**(5), 3181 (2023)
27. Wu, H., Lei, R., Peng, Y., Gao, L.: AagNet: a graph neural network towards multi-task machining feature recognition. *Robot. Comput.-Integr. Manuf.* **86**, 102661 (2024)
28. Zhang, S., Guan, Z., Jiang, H., Wang, X., Tan, P.: BrepMFR: enhancing machining feature recognition in b-rep models through deep learning and domain adaptation. *Comput. Aided Geometric Des.* **111**, 102318 (2024)
29. Zhang, Z., Jaiswal, P., Rai, R.: FeatureNet: machining feature recognition based on 3D convolution neural network. *Comput. Aided Des.* **101**, 12–22 (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Efficient Classification of Histopathology Images Using Highly Imbalanced Data

Mohammad Iqbal Nouyed¹(✉), Mary-Anne Hartley², Gianfranco Doretto¹,
and Donald A. Adjeroh¹

¹ West Virginia University, Morgantown, WV 26506, USA

monouyed@mix.wvu.edu, {gianfranco.doretto,donald.adjeroh}@mail.wvu.edu

² Yale University School of Medicine, New Haven, CT 06510, USA

mary-anne.hartley@yale.edu

Abstract. This work addresses how to efficiently classify challenging histopathology images, such as gigapixel whole-slide images for cancer diagnostics with image-level annotation. We use images with annotated tumor regions to identify a set of tumor patches and a set of benign patches in a cancerous slide. Due to the variable nature of region of interest the tumor positive regions may refer to an extreme minority of the pixels. This creates an important problem during patch-level classification, where the majority of patches from an image labeled as ‘cancerous’ are actually tumor-free. This problem is different from semantic segmentation which associates a label to every pixel in an image, because after patch extraction we are only dealing with patch-level labels. Most existing approaches address the data imbalance issue by mitigating the data shortage in minority classes in order to prevent the model from being dominated by the majority classes. These methods include data re-sampling, loss re-weighting, margin modification, and data augmentation. In this work, we mitigate the patch-level class imbalance problem by taking a divide-and-conquer approach. First, we partition the data into sub-groups, and define three separate classification sub-problems based on these data partitions. Then, using an information-theoretic cluster-based sampling of deep image patch features, we sample discriminative patches from the sub-groups. Using these sampled patches, we build corresponding deep models to solve the new classification sub-problems. Finally, we integrate information learned from the respective models to make a final decision on the patches. Our result shows that the proposed approach can perform competitively using a very low percentage of the available patches in a given whole-slide image.

Keywords: histopathology image · data imbalance · patch classification

1 Introduction

Whole-slide images (WSIs) are a rich source of information in digital histology, where tissue sections are scanned at gigapixel scale at various microscopic

magnification levels [20, 41]. However, the size and number of these images pose challenges for machine learning models. Firstly, the gigapixel resolution creates memory constraints necessitating input fragmentation. Secondly, annotations of the tumor regions may constitute a very tiny portion of the entire WSI which can create a large class imbalance in the training data. In recent years, deep models like CNNs and Transformer-based weakly supervised learning methods such as multiple-instance learning (MIL) have shown promising results in gigapixel whole slide image classification with varied sizes. In this approach, WSIs are divided into small image tiles or patches and then aggregated in later stages to make prediction using a classifier [21, 32, 40]. MIL treats each WSI as a bag containing multiple instances. If any instance of a WSI is disease-positive then the whole bag (WSI) is labeled as disease-positive. An aggregator classifier is used on the instance-level predictions to get the final image level prediction.

Real-world datasets often display long-tailed or imbalanced class distributions [7, 10, 12, 39]. Common approaches to handling data imbalance work by mitigating the data shortage in minority class by data augmentation [6, 17, 25], margin modification [4], loss re-weighting [9, 30, 36], and data re-sampling [1, 5, 24, 37]. Though these methods have performed well on imbalanced natural image data, they may not be as effective for WSIs. This is because, in the MIL classifier, the WSI is represented as a bag of image tiles of variable sizes [38]. However, since the area of the image that actually contains tumor in a WSI can be very small, it means that a majority of tiles in an image weakly labeled 'cancerous' actually do not contain tumor, effectively mislabeling ($>80\%$) of the tiles [18].

To address these challenges, we propose a patch-level classification method that utilizes cluster-based sampling strategy to solve the imbalance problem between tumor and benign class patches and also provides an efficient histopathology image classification framework for resource-constrained scenarios. The main contributions of this work are as follows:

- A group based training approach where we divide the data into three specific sets which help us to decompose the original problem into three sub-problems. Each sub-problem focuses on discriminating between specific binary classification problems and, when combined, solves the original classification challenge effectively.
- A z-score-based stratified sampling on clustered image patches of the three focus data groups, which allows us to sample most of the patch texture variety by selecting patches from all the distance-based intervals from cluster centroid.
- A learning based information integration from the three sub-problems to obtain the final image level predictions.

2 Related Work

2.1 Multiple Instance Learning (MIL) for WSI Classification

A typical MIL method for WSI classification consists of two stages. First, features are extracted from each instance, and then these instance features are

aggregated to obtain a bag-level feature. Then, an image (bag) level classifier is trained using the bag-level features and their corresponding labels. Lin et al. [22] proposed a model-agnostic framework called CIMIL to improve existing MIL models by using a counterfactual inference-based subbag evaluation method and a hierarchical instance searching strategy to help search reliable instances and obtain their accurate pseudo-labels. Qu et al. [28] proposed a feature distribution-guided MIL framework called DGMIL, for both WSI classification and positive patch localization. Shi et al. [35] proposed a loss based attention mechanism, which simultaneously learns instance weights and predictions, and bag predictions for deep MIL. Qu et al. [29] proposed an end-to-end weakly supervised knowledge distillation framework called WENO for WSI classification. Li et al. [21] proposed a deep MIL model, called DSMIL, which jointly learns a patch (instance) and an image (bag) classifier, using a two-stream architecture. Zhang et al. [40] proposed to virtually enlarge the number of bags by introducing the concept of pseudobags, on which a double-tier MIL framework, called DFTD-MIL, is built to effectively use the intrinsic feature. Kong et al. [18] presented an end-to-end CNN model called the Zoom-In network that uses hierarchical attention sampling to classify gigapixel pathology images with minority-pixel cancer metastases from the CAMELYON16 dataset. Sharma et al. [33] proposed an end-to-end framework named Cluster-to-Conquer (C2C) that clusters the patches from a WSI into k -groups, samples k' patches from each group for training, and uses an adaptive attention mechanism for slide-level prediction the patches. Campanella et al. [3] presented a deep learning system based on multiple instances of learning that uses only the diagnoses reported as labels for training, thereby avoiding expensive and time-consuming pixel-wise manual annotations. Lu et al. [23] reported an interpretable weakly supervised deep-learning method called CLAM that uses attention-based learning to identify sub-regions of high diagnostic value to accurately classify whole slide images. Nouyed et al. [26] addressed the challenge of high resolution image classification using a discriminative patch selection approach where they embed their patch selection approach inside a novel classification framework which can support the use of different off-the-shelf deep models.

While all the works mentioned above focus on solving the problem of patch-level label corruption from weakly assigned labels at the image level, they do not address the frequent issue of patch class imbalance, where the region of interest (ROI) that defines the label occupies a super minority of the image pixel space. Pawlowski et al. [27] investigated the performance of CNNs for minority-pixel image classification tasks and their results show that by using a training dataset limited in size, CNNs fail to generalize well because of the low ROI-to-image ratio. Usually, the object associated with the label occupies a dominant portion of the image. However, in histopathology image classification such as gigapixel whole-slide image classification, there could be datasets where only a very tiny fraction of the image informs the positive label.

2.2 Long-Tailed Histopathology Image Classification

Long-tailed classification is a well-known research topic in machine learning where the objective is to solve the data imbalance problem [15, 19]. Under-sampling [1, 11, 24] and over-sampling [31, 34] are common solutions with known trade-offs between bias and accuracy. While over-sampling can lead to overfitting of the minority class [5], under-sampling has the potential of information loss about the majority class [24]. We can also apply data augmentation to amplify the minority classes [6, 17]. Another category of data balancing is called loss re-weighting, in which the loss function is modified to increase weight on the minority class samples and decrease weight on the majority class samples [9, 14, 16]. But research has shown that loss re-weighting can be ineffective when the datasets are separable [2].

3 Method

In MIL, a group of training samples is considered as a bag containing multiple instances. Each bag has a bag label that is positive if the bag contains at least one positive instance and negative if it contains no positive instance. The instance-level labels are unknown. In the case of binary classification, let $B = \{(x_1, y_1), \dots, (x_n, y_n)\}$ be a bag where $x_i \in \mathcal{X}$ are instances with labels $y_i \in \{0, 1\}$, the label of B is given by

$$c(B) = \begin{cases} 0, & \text{iff } \sum y_i = 0 \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

First, the image is split into $N \times N$ instances of equal size. We consider the instances from the same image as in the same bag. The main components of our method can be divided into four parts, (1) A divide-and-conquer approach is taken by splitting training data into 3 patch sets using pseudo-labeling and ROI; (2) partitioning of the cancer classification problem into smaller sub-problems based on tumor annotation and source of tissue sample; and (3) Integration of the patch level results using patch level pooling at the feature and prediction levels, followed by activation function and dimensionality reduction (if needed), (4) A threshold percentage of tumor patch per WSI is used to determine the final patch level prediction. Figure 1 provides an overview of the proposed framework, while Fig. 2 shows a sample WSI with annotated tumor regions.

3.1 Partition-Based Approach to WSI Analysis

Based on the annotation provided in the dataset, we categorize the patches into three different types: 1) Set of tumor patches, denoted as A , so A can be defined as $A = \{(x, y_p) | y_p = 1, x \in W, y_w = 1\}$, where x is an image patch, y_p is patch label, W is an image, and y_w is image label; 2) Set of benign patches that belong to WSIs labeled as cancerous, denoted as set B , so $B = \{(x, y_p) | y_p = 0, x \in$

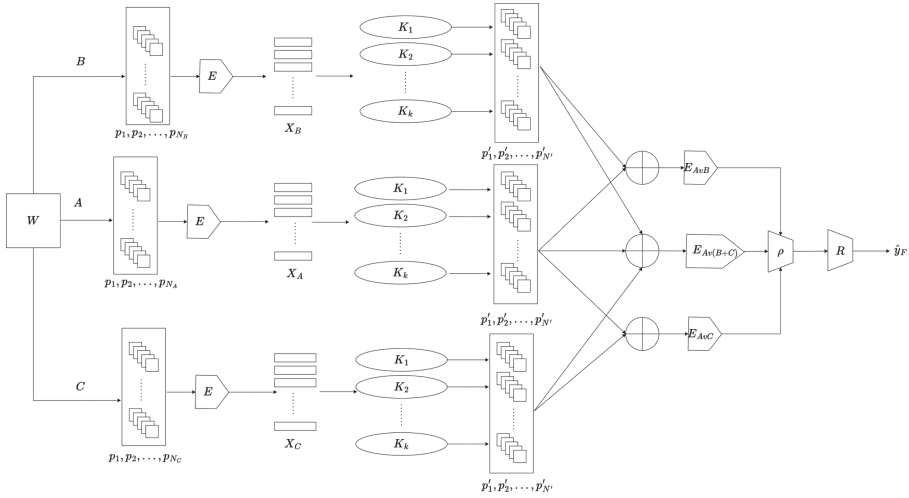


Fig. 1. Overview of the proposed framework. At the first stage, all patches of WSIs are extracted using a pre-trained model E . Then based on available annotation train set data is categorized into 3 data sub-sets. Feature set X_A, X_B, X_C are extracted from each corresponding set. On each set clustering K is performed and then z-score based cluster sampling strategy is applied. Then 3 different models E_{AvB}, E_{AvC} and $E_{Av(B+C)}$ are fine-tuned using the sampled patches $\{p'_1, p'_2, \dots, p'_N\}$ to train the binary classification models $E_{AvB}, E_{AvC}, E_{Av(B+C)}$. From these, the feature or aggregation information is passed to the aggregation function $\rho(.)$ for patch-level aggregation. And, these aggregated information is used for final patch-level decision fusion using the final R classifier.

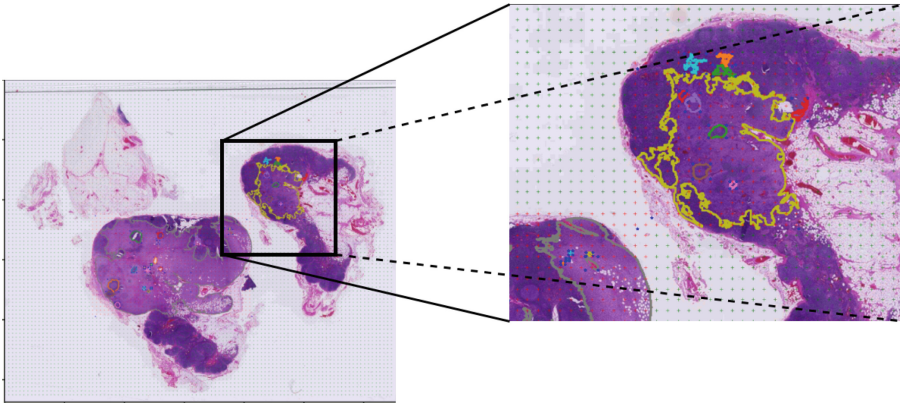


Fig. 2. Sample WSI, with annotation. Zoomed in section includes annotated regions in different colors, also, the '+' signs indicate the boundaries of the extracted 256×256 size patches.

$W, y_w = 1$ }; (Note that, set B does not indicate a misclassification, a doctor’s misdiagnosis, nor a system’s misdetection. It is simply the set of patches that are extracted outside the annotated tumor regions of the WSI.); and 3) Set of benign patches that belong to WSIs labeled as benign, denoted as set C, so $C = \{(x, y_p) | y_p = 0, x \in W, y_w = 0\}$. The assumption is that the benign patches belonging to a cancerous WSI may contain additional information that can help the model learn to better discriminate between tumor and tumor-free patches. Based on these three data partitions, we can now decompose the original problem into three different binary classification sub-problems: 1) A versus B, 2) A versus C and 3) A versus (B+C). We then train three different classification models of the same architecture for each of the patch-level classification sub-problems.

3.2 Information Theoretic Cluster-Based Sampling

Algorithm 1. Information-theoretic cluster-based patch sampling algorithm

Require: $X, K, |A|$

Ensure: P

```

1: for  $K_1, K_2, \dots, K_k$  do
2:   for  $x \in X$  do
3:      $d \leftarrow JSD(K_k, x); D(k) \leftarrow D(k) \cup d$ 
4:   end for
5:    $D_\sigma(k) \leftarrow \sigma(D(k)); D_\mu(k) \leftarrow \mu(D(k))$ 
6:   for  $d \in D(k)$  do
7:      $z \leftarrow \frac{d - D_\mu(k)}{D_\sigma(k)}; Z(k) \leftarrow Z(k) \cup z$ 
8:   end for
9: end for
10: for  $K_1, K_2, \dots, K_k$  do
11:    $S_T = |A| * ||D_\mu(k)||$ 
12:   for  $x \in X$  do
13:      $d \leftarrow JSD(K_k, x); z \leftarrow \frac{d - D_\mu(k)}{D_\sigma(k)}; i \leftarrow GetInterval(z); X'(i) \leftarrow x$ 
14:   end for
15:    $s_i = \lfloor S_T / |X'| \rfloor$ 
16:   while true do
17:     for  $i \in X'$  do
18:        $\rho \leftarrow RandomSample(X'(i)); P \leftarrow P \cup \rho; S_T \leftarrow S_T - |\rho|$ 
19:     end for
20:      $Update(X'); s_i = \lfloor S_T / |X'| \rfloor$ 
21:     if  $S_T \leq 0$  or  $s_i \leq 0$  then
22:       break
23:     end if
24:   end while
25: end for

```

Because of the partition of the patch sets, we now have a clearer understanding of the class imbalance between the benign and tumor patches. Typically for a

dataset $|A| \ll |B|$ and $A \ll |C|$. For this reason, we apply a sampling approach to reduce the class imbalance among set A , B and, C . First using a pre-trained model, we extract features from all patches. Let $X \leftarrow f(B, \theta)$ or $X \leftarrow f(C, \theta)$, where $f(\cdot, \theta)$ is a feature extractor using the trained parameters θ . We use a parametric clustering method to cluster each of the sets into k different clusters, such that, K_k is the k -th cluster centroid. Given a set of patch features (x_1, x_2, \dots, x_n) , where each patch has been converted to a d -dimensional real vector, parametric clustering such as k-means aims to partition the n patches into k cluster sets ($k \leq n$) $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$ so as to iteratively minimize the within-cluster sum of square errors to reach the local minima or optimum. The objective can be defined as :

$$\underset{\mathbf{S}}{\operatorname{argmin}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|^2 \quad (2)$$

where, μ_i is the mean or centroid of the points in S_i , $\mu_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x}$. We use these k clusters to perform a systematic sampling on the patches such that we can create balanced sets that are not dominated by the minority class. Algorithm 1 shows our procedure for this information-theoretic cluster-based sampling to in generating balanced patch sets. Based on the available tumor patches, we sample equal number of patches from each of the k cluster sets, if B' and C' are the new sampled sets such that $B' \subset B$ and $C' \subset C$ then $|B'| = |A|$, $|C'| = |A|$. Denote $P = B'$ or C' . During clustering we take a stratified random sampling approach based on the Euclidean distance from the cluster centroid in order to maximize the intra cluster variance among the clusters by sampling in such a way that P contains samples from all z-score intervals $X'(i)$. For a given patch in a cluster, we represent its computed features as a probability distribution. The cluster centroid is also represented similarly. We then use an information-theoretic divergence measure, namely the Jensen-Shannon divergence (JSD), to evaluate the dispersion between the patch, and its cluster centroid. For two probability distributions p_1 and p_2 , the Jensen-Shannon divergence [8] is given by:

$$JSD(p_1, p_2) = \frac{1}{2} D(p_1 || q) + \frac{1}{2} D(p_2 || q) \quad (3)$$

where $q = \frac{1}{2}(p_1 + p_2)$, and $D(p_1 || q)$ is the Kullback-Leibler (KL) divergence [8] between two distributions, given by:

$$D(p_1 || q) = \sum_{c=1}^{|C|} p_1(c) \log \left(\frac{p_1(c)}{q(c)} \right) \quad (4)$$

where C is the number of distinct intervals used in the representation. For each cluster we divide the distribution into intervals based on the z-scores. Then for each patch we calculate the z-score of its dispersion from the centroid $z \leftarrow \frac{d - D_{\mu}(k)}{D_{\sigma}(k)}$. Based on this, we make sure we uniformly sample from each z-score interval as much as possible so that we can have representation of all possible patch texture variances as much as possible from each cluster, while keeping the

total sample size within S_T where $S_T = |A|*||D_\mu(k)||$. Essentially, S_T is the value we get by multiplying the expected total size of the sampled set with normalized mean of all centroid dispersions (or centroid distances). A pseudo-code based description is provided in Algorithm 1.

3.3 Instance Level Learning

The instance-level models encode patches to a d -dimensional embedding, $f(\mathbf{x}, \theta) : \mathbf{x} \rightarrow \mathbf{h}$ where θ is the set of training parameters. During the training, we use the cross entropy loss on the instance-level labels and prediction of the selected instance to update the classifier's parameters. The loss function for the classifier is define as follows:

$$L = - \sum_j y_j \log \hat{y}_j + (1 - y_j) \log (1 - \hat{y}_j), \quad (5)$$

where y_j is the instance-level label. Using the partitioned datasets A, B and C we train three different binary classification models, that learns to discriminate between A vs. B, A vs. C, and A vs. (B+C), respectively. The objective here is to divide the problem space into sub-problems that discriminate between tumor and benign regions within same tissue image (AvB); between tumor and benign regions of other tissue images (AvC), and tumor and benign regions of both same and other tissue image (Av(B+C)). The assumption is that the aggregated feature representations obtained from these expert binary classification models will be more informative for the final prediction. See Fig. 1.

3.4 Integrating Information from Problem Decompositions

We investigate information integration from the sub-problems in 5 different ways: (M0) Majority vote based on the fine-tuned deep model predictions: Let, $\hat{Y}_{AvB} = \{\hat{y}_i | f_{AvB}^\theta(x_i, y_i) \rightarrow \hat{y}_i\}$, $\hat{Y}_{AvC} = \{\hat{y}_i | f_{AvC}^\theta(x_i, y_i) \rightarrow \hat{y}_i\}$ and $\hat{Y}_{Av(B+C)} = \{\hat{y}_i | f_{Av(B+C)}^\theta(x_i, y_i) \rightarrow \hat{y}_i\}$, are the set of instance-level predictions obtained from models trained on the AvB, AvC, and Av(B+C) datasets. Here f_{AvB} , f_{AvC} , $f_{Av(B+C)}$ are the binary classification models trained on some parameters θ , and \hat{y}_i is the predicted label of the i -th instance. Then we perform a simple majority vote count based on the number of positive predictions to obtain \hat{y}_F , the fused label; (M1) Learning-based fusion using Softmax: Let, $S_{AvB} = \{\sigma_i | \mathcal{S}(f_{AvB}^\theta(x_i, y_i)) \rightarrow \sigma_i\}$, $S_{AvC} = \{\sigma_i | \mathcal{S}(f_{AvC}^\theta(x_i, y_i)) \rightarrow \sigma_i\}$ and $S_{Av(B+C)} = \{\sigma_i | \mathcal{S}(f_{Av(B+C)}^\theta(x_i, y_i)) \rightarrow \sigma_i\}$ be a set of instance-level probability distributions, where $\mathcal{S}(\cdot)$ is the softmax function. These instance level probability distributions are concatenated to x and passed to a classifier; (M2) learning-based fusion using feature concatenation followed by dimensionality reduction: Let, $X_{AvB} = \{x_i | E_{AvB}^\theta(x_i)\}$, $X_{AvC} = \{x_i | E_{AvC}^\theta(x_i) \rightarrow x_i\}$ and $X_{Av(B+C)} = \{x_i | E_{Av(B+C)}^\theta(x_i) \rightarrow x_i\}$, be the set of feature representations obtained from the trained feature encoders E_{AvB}^θ , E_{AvC}^θ , $E_{Av(B+C)}^\theta$. These features are concatenated to x and then passed to a dimensionality reduction function $PCA(\cdot)$ followed by a classifier; (M3) Learning-based fusion by applying

dimensionality reduction on individual features and then concatenation. Similar to (M2), but here first $PCA(\cdot)$ is applied on individual feature representation and then the reduced features are concatenated; and (M4) Instance level pooling of learned features followed by activation functions and then applying a classifier. Average pooling method on the instance level features is applied to obtain an aggregated patch level representation. This aggregated representation is then passed to GeLU ($G(z)$) function followed by a classifier, where $G(z)$ is defined as follows.

$$G(z) = 0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)]) \quad (6)$$

Algorithm 2 shows our proposed procedures for integrating information obtained from solving the three sub-problems.

Algorithm 2. Information integration from the problem decomposition

Require: $X_{AvB}, X_{AvC}, X_{Av(B+C)}, \hat{Y}_{AvB}, \hat{Y}_{AvC}, \hat{Y}_{Av(B+C)}, S_{AvB}, S_{AvC}, S_{Av(B+C)}, Y, m$

Ensure: \hat{y}_F

```

1: if  $m = 0$  then
2:   for  $\hat{y}_1 \in \hat{Y}_{AvB}, \hat{y}_2 \in \hat{Y}_{AvC}, \hat{y}_3 \in \hat{Y}_{Av(B+C)}$  do
3:     if  $\hat{y}_1 + \hat{y}_2 + \hat{y}_3 \geq 2$  then return 1 else return 0 end if
4:   end for
5: else if  $m = 1$  then
6:   for  $\sigma_1 \in S_{AvB}, \sigma_2 \in S_{AvC}, \sigma_3 \in S_{Av(B+C)}, y \in Y$  do
7:      $x \leftarrow [\sigma_1; \sigma_2; \sigma_3]; \hat{y}_F \leftarrow \text{RandomForest}(x, y);$  return  $\hat{y}_F$ 
8:   end for
9: else if  $m = 2$  then
10:  for  $x_1 \in X_{AvB}, x_2 \in X_{AvC}, x_3 \in X_{Av(B+C)}, y \in Y$  do
11:     $x \leftarrow [x_1; x_2; x_3]; x' \leftarrow PCA(x); \hat{y}_F \leftarrow \text{RandomForest}(x', y);$  return  $\hat{y}_F$ 
12:  end for
13: else if  $m = 3$  then
14:  for  $x_1 \in X_{AvB}, x_2 \in X_{AvC}, x_3 \in X_{Av(B+C)}, y \in Y$  do
15:     $x'_1 \leftarrow PCA(x_1), x'_2 \leftarrow PCA(x_2); x'_3 \leftarrow PCA(x_3)$ 
16:     $x' \leftarrow [x'_1; x'_2; x'_3]; \hat{y}_F \leftarrow \text{RandomForest}(x', y);$  return  $\hat{y}_F$ 
17:  end for
18: else if  $m = 4$  then
19:  for  $x_1 \in X_{AvB}, x_2 \in X_{AvC}, x_3 \in X_{Av(B+C)}, y \in Y$  do
20:     $x'_p \leftarrow \text{AvgPool}([x'_1; x'_2; x'_3]); x' \leftarrow GeLU(x'_p)$ 
21:     $\hat{y}_F \leftarrow \text{RandomForest}(x', y);$  return  $\hat{y}_F$ 
22:  end for
23: end if

```

4 Experiments and Results

4.1 Database

We use the publicly available CAMELYON16 dataset for breast cancer metastasis detection. It has a total of 399 WSIs, with 270 WSIs in training and 129

Table 1. CAMELYON16 dataset details with info on 256×256 size patches extracted at $10\times$ magnification level.

		Train	Test
Image level (WSI)	Total number	270	129
	Positives (number, %)	111 (41%)	49 (31%)
	Negatives (number, %)	159 (59%)	80 (62%)
	Avg. area of ROI (pixels, %)	444,770 (0.003%)	653,670 (0.005%)
	Max area of ROI (pixels, %)	91,418,800 (0.8%)	332,954,015 (2.8%)
	Min area of ROI (pixels, %)	10 (0.000%)	0 (0.000%)
Patch level	Total number	4,612,746	2,026,538
	Positives (number, %)	38,052 (0.8%)	31,536 (1.56%)
	Negatives (number, %)	4,574,694 (99%)	1,995,131(98%)
Patches/Image (PPI)	Average	17083	15710
	Max	22,787	20,906
	Min	1,461	3,093

WSIs in test set. Out of the 270 training images, 111 are tumor WSIs, whose tumor annotation is also provided. For our work, patches of size 256×256 at $10\times$ magnification were extracted. Table 1 provides both slide-level and patch-level database details. Figure 2 shows a sample WSI with tumor annotation.

4.2 Architecture and Hardware

For all models, we used ResNet-18 [13] with a $l = 512$ feature representation which was then clustered using k-means with l_2 -normalization. The model was implemented with PyTorch and trained on a single RTX1080 GPU. The models are trained using an SGD optimizer with a batch size of 512 and a learning rate of $1e - 4$ for 10 epochs.

4.3 Patch Labeling

To partition the patches into groups (A, B, and C), first we find the bounding box around the tumor polygons provided by the CAMELYON16 dataset. After that, for each patch of a WSI, we detect if there is any overlap between the polygon bounding box and the patch coordinates, if there is an overlap we calculate the area of overlapping rectangles using the following formula:

$$(\min(x_2, p_2) - \max(x_1, p_1)) * (\min(y_2, q_2) - \max(y_1, q_1)) \quad (7)$$

where $(x_1, y_1), (x_2, y_2)$ are the polygon bounding box, and $(p_1, q_1), (p_2, q_2)$ are the patch coordinates. We use an overlap threshold to decide whether to assign the patch of a tumor positive WSI in A set, or in the B set, and, if the WSI is tumor negative we put the patches in C set.

Table 2. Distribution of patches after partitioning into the 3 groups and after applying the clustering based sampling algorithm to create balanced sets.

		A	B	C	Total
Unbalanced data	Train	30,442	1,505,762	2,153,994	3,690,198
	Val	7,610	376,440	538,498	922,548
	Total	38,052	1,882,202	2,692,492	4,612,746
Balanced data	Train (number, %)	30,442 (100%)	30,440 (2%)	30,440 (1.4%)	91,322 (2.4%)
	Val (number, %)	7,610 (100%)	7,609 (2%)	7,609 (1.4%)	22,828 (2.4%)
	Total (number, %)	38,052 (100%)	38,049 (2%)	38,049 (1.4%)	114,150 (2.4%)

4.4 Patch Sampling

We apply a K-means clustering algorithm, with $k = 10$, on the pre-trained ResNet18 [13] features of the unbalanced training set. Now to sample from the B set and C set patches, equal to the size of A set, we use the Euclidean distance from centroid feature to patch feature. The z-score intervals span from -3 to 15, and patches are sampled from within these intervals. Table 2 provides the details on both unbalanced and balanced training and validation datasets. We can observe that for the training and validation set we tried to keep the A set patches as much as possible so that we don't lose any information regarding tumor presence in the slides. The contributions of the cluster-based sampling strategy or the z-score based sampling strategy were visible, when we compared its performance with just random sampling once the sub-groups of patches are formed. We have found due to the Gaussian nature of the random sampling algorithm most of the patches were similar and does not represent all the variable texture patches within the centroid. This motivated us to take the z-score based sampling approach so that we can properly sample representations from all ranges of variability within a cluster.

4.5 Efficiency

From Table 2, we can observe that we have used 100% of all A set patches for the training and validation set construction, but reduced the majority classes (namely, class B and class C) down to 2% and 1.4% of the original dataset, respectively, in order to match with the minority class. Since during training time these patches are processed sequentially, the time that can be saved can be estimated as $\mathcal{O}(B/U)$, where B is the total size of the balanced data, and U is the total size of the unbalanced data. Thus, from the table, the proposed method will run about 50 times faster than working without the proposed sampling approach. Note that, we are estimating the efficiency gain based on the presence of the balancing step in the proposed framework. We also observe that, speed of convergence is another aspect of measuring the efficiency of the balancing approach which can further establish the efficacy of the balancing step.

Table 3. Cross validation result for the 3 models.

	AvB	AvC	Av(B+C)
	Top-1 Acc.	Top-1 Acc.	Top-1 Acc.
Avg. \pm Std.	0.894 ± 0.023	0.902 ± 0.022	0.897 ± 0.011

Table 4. Patch-level classification performance using the proposed models for information integrating from the problem decompositions.

Methods	Accuracy	AUC	Precision	Recall	F1-score
M0	0.833 ± 0.001	0.833 ± 0.001	0.764 ± 0.001	0.963 ± 0.001	0.852 ± 0.001
M1	0.980 ± 0.001	0.980 ± 0.001	0.964 ± 0.002	0.997 ± 0.001	0.980 ± 0.001
M2	0.989 ± 0.000	0.989 ± 0.001	0.978 ± 0.001	0.999 ± 0.000	0.988 ± 0.001
M3	0.988 ± 0.001	0.989 ± 0.001	0.978 ± 0.001	0.999 ± 0.000	0.988 ± 0.000
M4	0.987 ± 0.001	0.987 ± 0.004	0.975 ± 0.001	0.999 ± 0.000	0.987 ± 0.001

4.6 Evaluation

Using the balanced A, B and C sets, we create 5-fold cross validation sets (80–20 partition). We evaluated performances of the models trained on AvB, AvC and Av(B+C) models, individually and also using feature aggregated decision fusion approaches. Table 3 shows the individual cross-validation performance of the 3 binary classification models on the balanced datasets. The average top-1 accuracies are 0.894 ± 0.023 , 0.902 ± 0.022 and 0.897 ± 0.011 showing strong patch level performance on the individual partitioned data sub-sets. This is using a relatively weak ResNet backbone architecture (ResNet-18). Table 4 shows the performance of different feature aggregation and decision fusion strategies using the combined folds from the 3 partitioned datasets. This makes the folds harder to predict because they include samples from all A, B, and C sets. Even after that we can see that feature concatenation followed by PCA and RF classification (M2) shows a strong performance of top-1 accuracy 0.989 along with high precision (0.978 ± 0.001), recall (0.999) and F1-score (0.988 ± 0.0005). The second best method utilizes PCA on deep features following by concatenation of dimensionality reduced features before classification (M3), has almost the same performance as M2. In fact, except for majority vote approach (M0), all 4 learning-based approaches show strong patch-level classification performance on the validation set. We believe this is indicative of the efficacy of our partitioning, sampling, and information integration from the three problem decompositions. Still the work has to show good performance on the test set also, which is much more challenging because we have to infer initial sub-divisions, followed by cluster sampling on an unseen data.

In Table 5 we provide comparative instance-level classification performance results with state-of-the-art methods. For our instance-level classification we used Area Under the Curve(AUC) and Top-1 accuracy as evaluation metrics

Table 5. Comparison of patch level classification performance with the state-of-the-art.

Methods	Accuracy	AUC
Loss-ABMIL [35]	0.803	0.848
CLAM-SB [23]	0.789	0.880
CLAM-MB [23]	0.799	0.878
DSMIL [21]	0.857	0.886
DSMIL+WENO [29]	0.901	0.930
DTFD-MIL [40]	0.870	0.893
DGMIL [28]	0.886	0.901
CLAM-SB+CIMIL [22]	0.921	0.943
M2 (Ours) ^a	0.989 ± 0	0.989 ± 0.001
M3 (Ours) ^a	0.988 ± 0.001	0.989 ± 0.001

^a Reported accuracy and AUC is based on validation data.

to compare with other methods. We chose Loss-ABMIL [35], CLAM-SB [23], CLAM-MB [23], DSMIL+WENO [29], CLAM-SB+CIMIL [22], DSMIL [21], and DTFD-MIL [40]. ABMIL, CLAM, DSMIL models are equipped with specific mechanisms that provide patch prediction, DGMIL is specifically tailored for patch classification. WENO and CIMIL are frameworks for boosting existing MIL models. It can be seen that even with a significantly reduced dataset we were able to achieve the best instance-level performance both in terms of accuracy and AUC. We note that, with very high data imbalance, AUC is a much more effective performance metric than accuracy. However, since we handled the large class imbalance problem as part of our proposed approach, we believe it is appropriate to then include accuracy for performance measurement.

5 Conclusion

In this work, we propose a patch-level classification method that utilizes a group based training approach. By compartmentalizing training into sub-groups, we decompose the original classification problem into smaller classification sub-problems. We then develop models to solve each smaller sub-problem. Information from these models are later aggregated using feature and decision fusion approaches leading to a superior classification result. Furthermore, the method also incorporates a cluster-based sampling strategy to solve the significant data imbalance problem between positive and negative classes while maintaining slide-level representation of all WSIs. This allows our approach to efficiently handle a large data source using limited computational resources. Strong patch-level performance in our cross-validation and data fusion experiments validates our claim. Future work should explore the transferable value of the patch level features for slide-level predictions and verify on test set data. More ablation studies is needed to further investigate the impact of various algorithmic parameters, e.g., the initial clustering, number of clusters, complexity of tumor patches, etc.

Acknowledgement. This work was supported in part by grants from the US National Science Foundation (Award #1920920 and #2125872).

References

1. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **106**, 249–259 (2018)
2. Byrd, J., Lipton, Z.: What is the effect of importance weighting in deep learning? In: Chaudhuri, K., Salakhutdinov, R. (eds.) *Proceedings of the 36th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, vol. 97, pp. 872–881. PMLR (2019)
3. Campanella, G., et al.: Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nat. Med.* **25**(8), 1301–1309 (2019)
4. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc. (2019)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
6. Chu, P., Bian, X., Liu, S., Ling, H.: Feature space augmentation for long-tailed data. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020*. LNCS, vol. 12374, pp. 694–710. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58526-6_41
7. Cong, C., Yang, Y., Liu, S., Pagnucco, M., Song, Y.: Imbalanced histopathology image classification using deep feature graph attention network. In: *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–4 (2022). <https://doi.org/10.1109/ISBI52829.2022.9761619>
8. Cover, T.M., Thomas, J.A.: *Elements of Information Theory* (2. ed.). Wiley (2006)
9. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
10. Deepak, S., Ameer, P.: Brain tumor categorization from imbalanced MRI dataset using weighted loss and deep feature fusion. *Neurocomputing* **520**, 94–102 (2023)
11. Drummond, C., Holte, R.C., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: *Workshop on Learning from Imbalanced Datasets II*, vol. 11, pp. 1–8 (2003)
12. Fotouhi, S., Asadi, S., Kattan, M.W.: A comprehensive data level analysis for cancer diagnosis on imbalanced data. *J. Biomed. Inform.* **90**, 103089 (2019)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
14. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. *Intell. Data Anal.* **6**(5), 429–449 (2002)
15. Karakoulas, G., Shawe-Taylor, J.: Optimizing classifiers for imbalanced training sets. In: Kearns, M., Solla, S., Cohn, D. (eds.) *Advances in Neural Information Processing Systems*, vol. 11. MIT Press (1998)

16. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans. Neural Networks Learn. Syst.* **29**(8), 3573–3587 (2018). <https://doi.org/10.1109/TNNLS.2017.2732482>
17. Kim, J., Jeong, J., Shin, J.: M2m: imbalanced classification via major-to-minor translation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
18. Kong, F., Henao, R.: Efficient classification of very large images with tiny objects. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2384–2394 (2022)
19. Kubat, M., Matwin, S., et al.: Addressing the curse of imbalanced training sets: one-sided selection. In: *ICML*, vol. 97, p. 179. Citeseer (1997)
20. Kumar, N., Gupta, R., Gupta, S.: Whole slide imaging (WSI) in pathology: current perspectives and future directions. *J. Digit. Imaging* **33**(4), 1034–1040 (2020)
21. Li, B., Li, Y., Eliceiri, K.W.: Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14318–14328 (2021)
22. Lin, W., Zhuang, Z., Yu, L., Wang, L.: Boosting multiple instance learning models for whole slide image classification: a model-agnostic framework based on counterfactual inference. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 4, pp. 3477–3485 (2024)
23. Lu, M.Y., Williamson, D.F., Chen, T.Y., Chen, R.J., Barbieri, M., Mahmood, F.: Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomed. Eng.* **5**(6), 555–570 (2021)
24. More, A.: Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint [arXiv:1608.06048](https://arxiv.org/abs/1608.06048)* (2016)
25. Mullick, S.S., Datta, S., Das, S.: Generative adversarial minority oversampling. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
26. Nouyed, M.I., Doretto, G., Adjero, D.A.: Efficient classification of very high resolution histopathological images. In: *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 3114–3121 (2022)
27. Pawlowski, N., Bhooshan, S., Ballas, N., Ciompi, F., Glocker, B., Drozdal, M.: Needles in haystacks: on classifying tiny objects in large images. *arXiv preprint [arXiv:1908.06037](https://arxiv.org/abs/1908.06037)* (2020)
28. Qu, L., Luo, X., Liu, S., Wang, M., Song, Z.: DGMIL: distribution guided multiple instance learning for whole slide image classification. In: Wang, L., Dou, Q., Fletcher, P.T., Speidel, S., Li, S. (eds.) *MICCAI 2022. LNCS*, vol. 13432, pp. 24–34. Springer, Cham (2022)
29. Qu, L., Luo, X., Wang, M., Song, Z.: Bi-directional weakly supervised knowledge distillation for whole slide image classification. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) *Advances in Neural Information Processing Systems*, vol. 35, pp. 15368–15381. Curran Associates, Inc. (2022)
30. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 80, pp. 4334–4343. PMLR (2018)
31. Sarafianos, N., Xu, X., Kakadiaris, I.A.: Deep imbalanced attribute classification using visual attention aggregation. In: Ferrari, V., Hebert, M., Sminchisescu, C.,

- Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11215, pp. 708–725. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_42
32. Shao, Z., et al.: Transmil: transformer based correlated multiple instance learning for whole slide image classification. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) *Advances in Neural Information Processing Systems*. vol. 34, pp. 2136–2147. Curran Associates, Inc. (2021)
33. Sharma, Y., Shrivastava, A., Ehsan, L., Moskaluk, C.A., Syed, S., Brown, D.: Cluster-to-conquer: A framework for end-to-end multi-instance learning for whole slide image classification. In: Heinrich, M., Dou, Q., de Bruijne, M., Lellmann, J., Schläfer, A., Ernst, F. (eds.) *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning. Proceedings of Machine Learning Research*, vol. 143, pp. 682–698. PMLR (2021)
34. Shen, L., Lin, Z., Huang, Q.: Relay backpropagation for effective learning of deep convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 467–482. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_29
35. Shi, X., Xing, F., Xie, Y., Zhang, Z., Cui, L., Yang, L.: Loss-based attention for deep multiple instance learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 5742–5749 (2020)
36. Tan, J., et al.: Equalization loss for long-tailed object recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
37. Wallace, B.C., Small, K., Brodley, C.E., Trikalinos, T.A.: Class imbalance, redux. In: 2011 IEEE 11th International Conference on Data Mining, pp. 754–763 (2011)
38. Wang, P., Han, K., Wei, X.S., Zhang, L., Wang, L.: Contrastive learning based hybrid networks for long-tailed image classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 943–952 (2021)
39. Wang, Q., et al.: WGAN-based synthetic minority over-sampling technique: improving semantic fine-grained classification for lung nodules in CT images. *IEEE Access* **7**, 18450–18463 (2019). <https://doi.org/10.1109/ACCESS.2019.2896409>
40. Zhang, H., et al.: DTFD-MIL: double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18802–18812 (2022)
41. Zuraw, A., Aeffner, F.: Whole-slide imaging, tissue image analysis, and artificial intelligence in veterinary pathology: An updated introduction and review. *Vet. Pathol.* **59**(1), 6–25 (2022)



GenFormer – Generated Images Are All You Need to Improve Robustness of Transformers on Small Datasets

Sven Oehri¹, Nikolas Ebert^{1,2(✉)}, Ahmed Abdullah¹, Didier Stricker²,
and Oliver Wasenmüller¹

¹ Mannheim University of Applied Sciences, Mannheim, Germany
{s.oehri,n.ebert,a.abdullah,o.wasenmueller}@hs-mannheim.de

² University of Kaiserslautern-Landau (RPTU), Kaiserslautern, Germany
didier.stricker@dfki.de

Abstract. Recent studies showcase the competitive accuracy of Vision Transformers (ViTs) in relation to Convolutional Neural Networks (CNNs), along with their remarkable robustness. However, ViTs demand a large amount of data to achieve adequate performance, which makes their application to small datasets challenging, falling behind CNNs. To overcome this, we propose GenFormer, a data augmentation strategy utilizing generated images, thereby improving transformer accuracy and robustness on small-scale image classification tasks. In our comprehensive evaluation we propose Tiny ImageNetV2, -R, and -A as new test set variants of Tiny ImageNet by transferring established ImageNet generalization and robustness benchmarks to the small-scale data domain. Similarly, we introduce MedMNIST-C and EuroSAT-C as corrupted test set variants of established fine-grained datasets in the medical and aerial domain. Through a series of experiments conducted on small datasets of various domains, including Tiny ImageNet, CIFAR, EuroSAT and MedMNIST datasets, we demonstrate the synergistic power of our method, in particular when combined with common train and test time augmentations, knowledge distillation, and architectural design choices. Additionally, we prove the effectiveness of our approach under challenging conditions with limited training data, demonstrating significant improvements in both accuracy and robustness, bridging the gap between CNNs and ViTs in the small-scale dataset domain.

Keywords: Robustness · Transformer · Classification · Generative

1 Introduction

Deep learning models, whether based on convolution or self-attention, achieve remarkable performances across a wide range of computer vision benchmarks.

S. Oehri and N. Ebert—Equal contribution.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78166-7_12.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15302, pp. 176–192, 2025.
https://doi.org/10.1007/978-3-031-78166-7_12

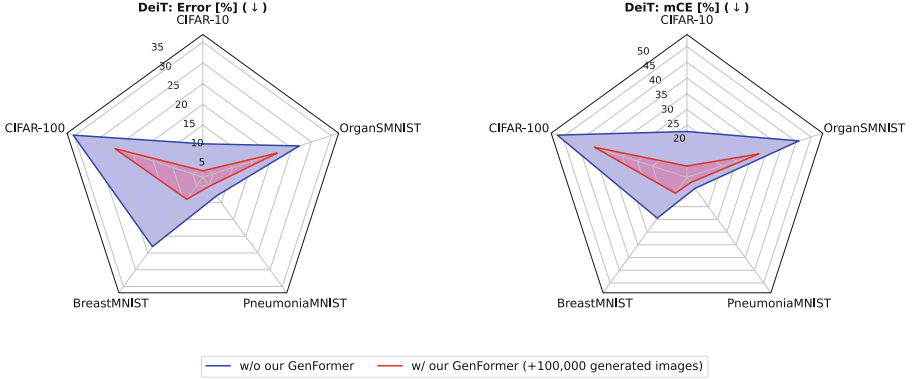


Fig. 1. Comparison of the error rate (left) and mean corruption error (right) of DeiT [46] on CIFAR [28] and parts of the MedMNIST [52] collection **with** and **without** our GenFormer. Lower error rates closer to the plot center are better.

Yet, numerous works demonstrate the vulnerability of modern architectures to adversarial perturbations [33], common corruptions [19], and domain shifts [18], a major challenge on the road to real-world applications. Recent research demonstrates the intrinsic robustness and generalization superiority of the transformer architecture compared to Convolutional Neural Networks (CNNs) [3, 36, 54]. However, a huge drawback of the Vision Transformer (ViT) [9] and its variants [10–12, 48] remains the demand for large-scale training data, due to its lack of an inductive bias which makes ViTs prone to overfitting when data is scarce.

Transfer learning is a commonly chosen approach to mitigate the problem of data scarcity by pre-training the model on an out-of-domain medium- or large-scale dataset [8] and subsequently fine-tuning on the target data. However, the viability of this approach is limited since specific domains (e.g. medical imaging [52]) lack large-scale datasets. Similar constraints apply to self-supervised approaches, such as masked image modeling (MIM) [14] which has proven to also rely on large amounts of data [50]. Thus, a plethora of works propose methods enabling ViTs to be trained on medium and small-sized datasets without the use of extra data [30, 32, 46]. However, exploiting the inherent robustness of transformers in data-limited scenarios remains a sparsely explored field.

In this work, we propose GenFormer, a data augmentation scheme enhancing the applicability of ViTs to small-scale datasets, utilizing their inherent robustness, by directly tackling data scarcity. We obtain additional information from the training data by expanding the real dataset with images from a generative model. Our GenFormer approach showcases accuracy and robustness improvements for a variety of transformer networks on downstream classification tasks, especially in domains with limited access to data (e.g. medical imaging [52]), as demonstrated by the results of DeiT [46] in Fig. 1. Since this work aims for a comprehensive investigation of our method’s impact on robustness in limited-data scenarios, we propose Tiny ImageNetV2, -R, and -A as new test sets of Tiny ImageNet by transferring established ImageNet [8] generalization [40] and robustness [18, 21] benchmarks to the small-scale data domain. Furthermore,

we demonstrate the straightforward applicability and synergistic effectiveness of our training scheme by combining it with existing approaches incorporating train and test time augmentation, knowledge distillation and architectural adaptations.

Our code and models are available on GitHub at <https://github.com/CeMOS-IS/GenFormer>. All test sets for evaluation can be found at <https://github.com/CeMOS-IS/Robust-Minisets>.

2 Related Works

Vision Transformers for Small Datasets. Recent efforts enable the use of Vision Transformers (ViTs) on small-scale datasets through novel training approaches. Strategies include advanced data augmentation [30, 46], knowledge distillation from CNNs [31, 46], and self-supervised objectives promoting spatial understanding [32]. Further approaches introduce architectural adaptations, such as adding convolutions to induce locality [34], or explicitly focusing on adapting the self-attention module to encourage a stronger focus on local [30] and meaningful [7, 34] information to prevent performance degradation on scarce datasets.

Robustness of Vision Transformers. In response to the success of the transformer architecture in terms of clean accuracy, many works study the performance of ViTs on robustness benchmarks [18, 19]. It becomes evident that transformers scale better with model size than CNNs and often surpass their convolutional counterparts when faced with corruptions [38, 45], adversarial attacks [33] and distribution shifts [54]. Training [38] and test time [56] augmentation methods utilizing AugMix [20] as well as attention enhancement techniques [7] have shown great improvement in robustness. Architectural changes such as position-based attention scaling [35] and Fully Attentional Networks (FANs) [57] further improve performance. Despite these advances, the use of robust transformers in data-constrained domains remains an emerging area of research.

Data Augmentation with Synthetic Images. Driven by the data hunger of modern deep learning models, exploiting information from synthetic data to expand training data for downstream tasks is gaining attraction in the field of computer vision. In the past, a plethora of works utilized synthetic datasets especially in domains requiring complex annotation such as human motion understanding [37] or semantic segmentation [42]. Early publications focusing on image classification are tailored to specific domains, lacking investigations on general applicability [22], partly not even deriving performance advantages [5].

Most general approaches focus on studying the effectiveness of substituting the real training set completely by generated data [39], thereby overlooking the potential of expanding the real dataset with synthetic images. For some approaches, this could be attributed to diminishing performance improvements when combining real and generated data [5]. More recent works employ pre-trained text-to-image models [41] to create synthetic classification datasets showing potential of being capable to replace [16, 43] and augment [2] general object-level datasets. Though, these approaches lack applicability to domains that deviate from common objects, such as medical imaging [52]. Some works apply generated images to modern transformer models [2, 16], others consider the effect of

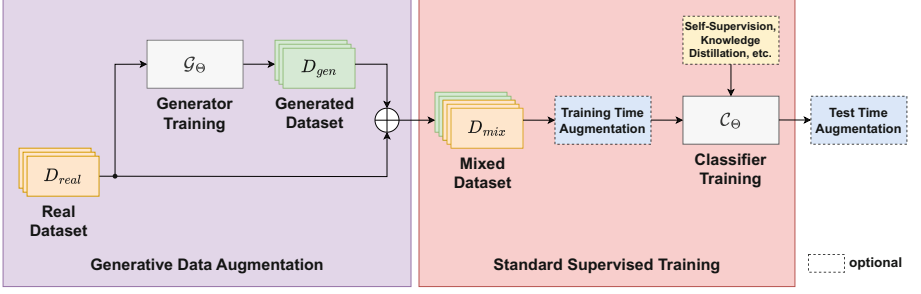


Fig. 2. The proposed GenFormer approach involves training a downstream-aware image generation model, \mathcal{G}_Θ , using real data D_{real} , then augmenting this dataset with generated data D_{gen} to create D_{mix} . Subsequently, \mathcal{C}_Θ is trained on D_{mix} for the classification task, with optional methods like data augmentation or knowledge distillation during training. \oplus denotes a concatenation.

synthetic images on domain shifts and adversarial robustness [13, 43]. However, to the best of our knowledge, no research explores the impact of data augmentation with generated images on the robustness of ViTs in limited-data scenarios.

In our work, we explore how generative data augmentation impacts transformer models’ robustness in small-scale data scenarios. Demonstrating that augmenting real datasets with images generated by a generative model improves accuracy and robustness against corruptions and domain shifts, our GenFormer approach proves to be versatile across domains. Additionally, we highlight its synergy with conventional methods like augmentation, knowledge distillation, and architectural adaptations, resulting in state-of-the-art performance across models trained from scratch.

3 Method

The aim of this work is to utilize the inherent robustness of the transformer architecture in limited-data scenarios. We therefore propose GenFormer, a generative data augmentation strategy, that alleviates the demand of transformers for large-scale datasets by tackling data scarcity directly, before common training schemes are applied. We accomplish this by exploiting information from generated images, which are created with knowledge about the label space of the downstream task. We refer to this as downstream-aware image generation. The exact procedure can be seen in Fig. 2.

Let $D_{real} = \{x_i^{real}, y_i^{real}\}_{i=1}^{N_{real}}$ be a training dataset where N_{real} denotes the number of real image-label pairs consisting of an image x^{real} and its corresponding class label y^{real} . Instead of simply following a standard supervised training strategy solely relying on the real training data, the proposed approach leverages additional information from the real dataset. This is achieved by a generative data augmentation utilizing a class-conditional downstream-aware image generation model \mathcal{G}_Θ learning the parameters Θ to approximate the true



Fig. 3. Real (left) and generated (right) sample pairs of corresponding classes (f.l.t.r): CIFAR-100 [28], BreastMNIST, PneumoniaMNIST, OrganSMNIST [52] and EuroSAT [17].

data distribution following the underlying objective $p_{\mathcal{G}}(x|y) \approx p^*(x|y)$. Here, x and y denote a random image with its corresponding label of the underlying data distribution whereas $p_{\mathcal{G}}$ and p^* denote the learned probability distribution of the generator and the true distribution respectively. The generator is then used to create N_{gen} images $x^{gen} = \mathcal{G}_{\Theta}(z, y_{gen})$ given a random noise vector z and a label y_{gen} as input. This results in an additional generated training dataset $D_{gen} = \{x_i^{gen}, y_i^{gen}\}_{i=1}^{N_{gen}}$. We then combine the real and generated data to receive $D_{mix} = \{D_{real}, D_{gen}\}$ which then is used to train a classifier \mathcal{C}_{Θ} on the downstream task. In the base setting we follow a standard supervised training.

As our GenFormer approach does not depend on a specific generator model, it is compatible with a wide range of image generation models. Based on downstream performance, we employ the EDM [24] diffusion model, which is showcased in Sect. 4.6. Since the generator model is trained on the real dataset, the amount of real training images also affects the generator model performance, which may become apparent in lower diversity of the generated images. However, in Sect. 4.3 we study the impact of the number of real images available for our method demonstrating consistent improvement through our approach. A big advantage of the proposed GenFormer approach results from its implementation before the standard downstream training. Thereby, our method allows the additional use of established techniques to increase accuracy and robustness, as can be seen in Fig. 2. In Sect. 4.2 we demonstrate the synergy of our approach with conventional data augmentation during train and test time as well as knowledge distillation and architectural modifications.

4 Evaluation

For our evaluation, we conduct extensive experiments involving a range of state-of-the-art Vision Transformers (ViTs) [10, 46–48] and convolutional neural networks (CNNs) [15, 49]. Unless otherwise mentioned, we employ EDM [24] as generator model \mathcal{G}_{Θ} to generate additional data, effectively expanding our datasets (see Sect. 4.6). Most training runs are carried out with the Tiny ImageNet [29] dataset. For robustness and generalization investigations, we also include our novel Tiny ImageNet versions of various established ImageNet benchmarks, Tiny ImageNetV2, -R, and -A, as well as Tiny ImageNet-C [19] in our studies.

In addition to Tiny ImageNet, we extend our investigation to the CIFAR-10 and CIFAR-100 datasets [28], using CIFAR-10-C and CIFAR-100-C [19]. We also consider the test set of CIFAR10.1 [40], which contains more challenging

samples. We analyzed three datasets of MedMNIST [52] from the medical field, which is known to have limited data. The aim of this study is to extend our evaluation beyond the domain of natural images. In addition, we investigated the impact of our method on the classification of aerial images using EuroSAT [17]. Examples of generated data are shown in Fig. 3. To maintain consistency in our evaluation, we report error rates rather than accuracy, with smaller values indicating better performance.

4.1 Robustness and Generalization Benchmarks

The evaluation of the robustness of neural networks plays a major role in real-world applications. More specifically, we focus on evaluating different types of robustness of ViTs on small datasets, which has not received much attention in research. To this end, we port the most common robustness and generalization benchmarks [18, 19, 21, 40] to Tiny ImageNet, MedMNIST [52] and EuroSAT [17].

We create Tiny ImageNetV2 to analyze the generalization ability by keeping all images of joint classes of Tiny ImageNet and ImageNetV2 [40]. In the same way, we introduce Tiny ImageNet-R to study the robustness of models when confronted with domain shifts, such as changes in the type of images (e.g. paintings, toys or graffiti). Lastly, we propose Tiny ImageNet-A. Based on ImageNet-A [21], we use all images from the original Tiny ImageNet validation set only keeping the images misclassified by ResNet-18 [15]. To further evaluate robustness, we utilize Tiny ImageNet-C of Hendrycks et al. [19], where the validation data from Tiny ImageNet is subjected to 15 different corruptions, each at five severity levels. Analogous to Tiny ImageNet-C, we introduce novel corrupted test set variants for established image classification benchmarks: EuroSAT-C, which targets aerial imagery, and MedMNIST-C, which focuses on the medical domain. To maintain the integrity of the medical data, we have excluded any weather-dependent corruptions (Snow, Frost, Fog). For evaluation, we use the Mean Corrupted Error (mCE). Further details on the test sets can be found in the Appendix (Sect. E).

4.2 Comparisons on Tiny ImageNet

In our first experiment, we perform a comparative analysis of our GenFormer approach in combination with established methods to improve the robustness of neural networks. Two transformer-based classifiers, the tiny versions of DeiT [46] (without distillation token) and PVT [47], are used for this comparison.

As shown in Table 1, our experiment demonstrates how our GenFormer approach seamlessly complements various techniques, resulting in notable improvements in both accuracy and robustness. As part of our comparative analysis, we employ well-established methods, including CutMix [53], Mixup [55] and AugMix [20] for data augmentation, Locality Guidance [31] for knowledge distillation, and MEMO [56] for test time augmentation (TTA). All of these methods are applied on top of our baseline, which is trained for 300 epochs on Tiny ImageNet. A detailed description of the training can be found in the Appendix (Sect. A).

Table 1. Analysis of our GenFormer in combination with established SOTA approaches on two Vision Transformers [46, 47] on Tiny ImageNet [29] and its robustness benchmarks [19]. 100,000 generated images are added to the real data.

Model	Train-Strategy		T-IN err.	T-INv2 err.	T-IN-R err.	T-IN-C mCE	T-IN-A err.
<i>DeiT-Ti</i> [46]	Baseline	w/o ours	50.3	68.0	92.5	80.6	80.6
		w/ ours	44.1 (-6.2)	65.3 (-2.7)	89.6 (-2.9)	77.7 (-2.9)	78.0 (-2.6)
	CutMix [53]	w/o ours	44.4	65.0	89.7	74.5	78.3
	+ Mixup [55]	w/ ours	38.5 (-5.9)	58.7 (-6.3)	86.4 (-3.3)	71.8 (-2.7)	74.4 (-3.9)
	AugMix [20]	w/o ours	40.4	61.3	88.4	72.8	76.2
		w/ ours	36.4 (-4.0)	57.8 (-3.5)	85.6 (-2.8)	71.0 (-1.8)	73.1 (-3.1)
	Locality	w/o ours	36.7	59.4	83.6	72.0	74.7
	Guidance [31]	w/ ours	36.2 (-0.5)	59.4 (± 0.0)	84.8 (+1.2)	70.7 (-1.3)	72.8 (-1.3)
		w/o ours	48.9	64.8	99.7*	78.1	99.7*
	MEMO [56]	w/ ours	43.1 (-5.8)	60.2 (-4.6)	99.7*	75.5 (-3.6)	99.8*
<i>PVT-T</i> [47]	Baseline	w/o ours	46.5	67.2	91.5	78.9	78.9
		w/ ours	42.8 (-3.7)	64.8 (-2.4)	87.4 (-4.1)	76.8 (-1.8)	77.7 (-1.2)
	CutMix [53]	w/o ours	41.9	64.6	87.8	74.1	76.8
	+ Mixup [55]	w/ ours	37.5 (-4.4)	60.1 (-4.5)	84.6 (-3.2)	70.1 (-4.0)	74.1 (-2.7)
	AugMix [20]	w/o ours	39.9	62.1	87.9	73.0	75.9
		w/ ours	36.4 (-3.5)	58.3 (-3.8)	86.2 (-1.7)	71.0 (-2.0)	73.6 (-2.3)
	Locality	w/o ours	36.3	58.5	84.9	72.3	72.2
	Guidance [31]	w/ ours	35.7 (-0.6)	58.1 (-0.4)	84.2 (-0.7)	71.0 (-1.3)	72.1 (-0.1)
		w/o ours	45.1	64.5	99.8*	76.6	99.6*
	MEMO [56]	w/ ours	42.3 (-2.8)	61.4 (-3.1)	99.7*	75.1 (-1.5)	99.6*

* leads to unstable results despite hyperparameter-tuning.

To expand the dataset with generated images, we use our method to create an additional 100,000 images (corresponds to 100% of the original dataset).

The results in Table 1 demonstrate the substantial benefits of incorporating additional generated data for both models. In the case of DeiT-Ti there is a significant reduction in the base error of -6.2 , which drops from the original 50.3 to 44.1. PVT-T initially shows a lower error even without additional data, but still benefits from an additional reduction of -3.7 to 42.8. Considering all presented robustness benchmarks, the errors of DeiT can be reduced by up to -2.9 , while the errors of PVT are reduced by up to -4.1 . These outcomes underline the significant potential of leveraging generated data. Combining our GenFormer approach with various data augmentation techniques such as Mixup and CutMix as commonly used with transformer classifiers and proposed by Touvron et al. [46], generated data is consistently shown to be beneficial for training. Notably, the combination of generated data with CutMix and Mixup results in a noteworthy improvement of -5.9 in Tiny ImageNet validation and an even more substantial reduction of -6.3 in Tiny ImageNetV2 for DeiT.

In addition to training time augmentation, we also explore the option of TTA, which is exemplified in our experiments through MEMO. When applied to our baseline networks, MEMO yields a moderate improvement, even without the inclusion of generated data. However, when we apply TTA to the baseline with generated data, we observe a significant higher enhancement. Furthermore,

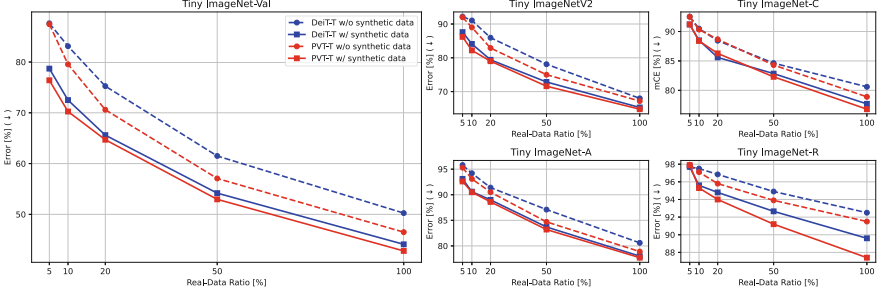


Fig. 4. Analysis of the influence of different amounts of training data on the accuracy and robustness of Vision Transformers [46, 47]. The networks are trained with {5, 10, 20, 50, 100}% of Tiny ImageNet [29]. We add 100,000 generated images (from a diffusion model trained on the same amount) to each train-set.

despite hyperparameter tuning, MEMO is not performing well on all benchmarks and can have a negative impact on the robustness of the models, especially in the case of Tiny ImageNet-R and -A. This emphasizes the advantages of generated data, as they are comparatively easy to integrate into the training pipeline and offer significant potential to increase accuracy and robustness.

When including Locality Guidance with a CNN teacher (ResNet-56) trained on the same data as the transformers, our GenFormer approach reduces the errors on clean data for both networks. However, we observe no improvement for Tiny ImageNetV2 and a slight degradation of +1.2 on Tiny ImageNet-R for DeiT. For all other benchmarks, we see an improvement in results. The performance gains achieved through generated data are moderate compared to training and test time augmentation, as our approach, like Locality Guidance, contributes to a better focus on more localized features. This is shown in the mean attention distances comparison presented in the Appendix (Sect. D).

4.3 Limited-Data Analysis on Tiny ImageNet

In our second experiment, we investigate the impact of limited data on the training and robustness of ViTs and the diffusion model used to generate the data. For this purpose, we divide the train set of Tiny ImageNet into four random subsets of 5%, 10%, 20% and 50%. In addition, we also run the experiment on the full set of data. In the first step, we trained the diffusion model for each of these subsets and generated a total of 100,000 images. In the second step, we carried out training sessions for PVT-T and DeiT-Ti, both with and without the inclusion of generated data. The results, including error rates on the validation set and robustness benchmarks, are presented in Fig. 4.

Congruent with previous findings, the accuracy of ViTs increases with the amount of real data. For instance, the error rate of DeiT, which starts at 87.6 when no generated images are added to a subset of 5%, decreases to 75.2 with 20% of the real data, and further drops to 50.3 when all 100,000 real images are utilized for training. Robustness across all benchmarks follows the same trend,

Table 2. Analysis of GenFormer for training multiple Vision Transformer [10, 35, 46–48, 57] and Convolutional Neural Networks [15, 49] on three datasets of the MedMNIST [1, 4, 26, 52] collection (medical domain) and on EuroSAT [17] (aerial domain) as well as their corresponding robustness benchmarks [19].

Model	B-MNIST(-C)		P-MNIST(-C)		OS-MNIST(-C)		EuroSAT(-C)		
	err.	mCE	err.	mCE	err.	mCE	err.	mCE	
Convolutional Neural Networks									
ResNet-18 [15]	w/o ours	17.9	19.8	4.2	8.6	17.4	30.4	1.5	24.6
	w/ ours	8.3 (-9.6)	15.6 (-4.2)	3.7 (-0.5)	7.7 (-0.9)	16.8 (-0.6)	26.5 (-3.9)	1.1 (-0.4)	23.1 (-1.5)
Conv-NeXtv2-F [49]	w/o ours	19.2	21.5	4.5	7.0	17.5	25.7	1.4	22.7
	w/ ours	9.0 (-10.2)	12.9 (-8.6)	4.2 (-0.3)	7.7 (+0.7)	16.2 (-1.3)	23.2 (-2.5)	0.9 (-0.5)	21.9 (-0.8)
Pure Vision Transformer									
DeiT-Ti [46]	w/o ours	23.1	24.5	8.0	12.8	26.9	45.9	2.4	27.7
	w/ ours	9.0 (-14.1)	14.7 (-9.8)	5.1 (-2.9)	10.5 (-2.3)	21.3 (-5.6)	32.5 (-13.4)	1.7 (-0.7)	26.9 (-0.8)
PVT-T [47]	w/o ours	22.4	23.9	13.1	17.8	20.3	34.4	1.9	26.2
	w/ ours	9.0 (-13.4)	13.9 (-10.0)	5.4 (-7.7)	10.0 (-7.8)	17.1 (-3.2)	24.7 (-9.7)	1.2 (-0.7)	25.4 (-0.8)
Hybrid Architectures									
PVTv2-B0 [48]	w/o ours	16.7	18.5	5.9	9.1	17.0	28.4	1.5	24.0
	w/ ours	9.0 (-7.7)	13.3 (-5.2)	4.5 (-1.4)	8.9 (-0.2)	16.2 (-0.8)	22.6 (-5.8)	1.0 (-0.5)	22.9 (-1.1)
ConViT-Ti [10]	w/o ours	16.0	22.2	6.3	11.8	19.6	35.3	2.6	28.0
	w/ ours	7.7 (-8.3)	12.0 (-10.2)	3.5 (-2.8)	6.8 (-5.0)	15.5 (-4.1)	22.4 (-12.9)	1.4 (-1.2)	25.7 (-2.3)
Robust Architectures									
RVT-Ti [35]	w/o ours	10.9	16.6	3.2	7.8	15.7	24.0	1.3	22.3
	w/ ours	7.7 (-3.2)	12.2 (-4.4)	3.7 (+0.5)	9.3 (+1.5)	16.0 (+0.3)	21.7 (-2.3)	1.0 (-0.3)	21.8 (-0.5)
FAN-T [57]	w/o ours	12.8	16.5	4.6	8.1	16.0	25.1	1.5	22.6
	w/ ours	8.3 (-4.5)	12.1 (-4.4)	3.4 (-1.2)	7.7 (-0.4)	15.2 (-0.8)	21.8 (-3.3)	1.2 (-0.3)	21.8 (-0.8)

showing consistent and steady improvement. When training with our method, there is a substantial reduction of the error and an increase of robustness, for both PVT and DeiT. Interestingly, even models with only 5% real data in conjunction with generated data achieve higher accuracy than models trained with 10% real images. This pattern is also evident when comparing models with 10% real data paired with generated images to those with 20% real data but without generated data. Nevertheless, using a small amount of real data (e.g. 5%, 10%, or 20%) in conjunction with a large amount of generated data is not as effective as training with a larger amount of real data (e.g. 50% or 100%).

4.4 Going Beyond Natural Images

In contrast to Tiny ImageNet and CIFAR, which consist mainly of natural images, other domains such as medical imaging suffer from a lack of data. It is not unusual for medical datasets to comprise only a few hundred to a few thousand images. Therefore, we assessed the MedMNIST [52] collection, which includes PneumoniaMNIST [26] with 5,856 images, OrganSMNIST [4] with 25,211 images, and BreastMNIST [1] with only 780 images. Additionally, we use EuroSAT [17] to classify aerial images, providing a more comprehensive evaluation across various fine-grained domains. In order to evaluate the robustness against common corruptions, the proposed MedMNIST-C and EuroSAT-C test sets are utilized.

In addition to pure ViTs such as DeiT-Ti and PVT-T, we include hybrid architectures [10, 48] combining self-attention with convolutional operations. We also investigate pure CNN architectures such as ConvNeXtv2-F [49] and ResNet-18 [15]. Furthermore, we analyze transformer architectures such as RVT-Ti [35]

Table 3. Analysis of GenFormer for training multiple Vision Transformers with [10, 35, 46–48, 57] and Convolutional Neural Networks [15, 49] on CIFAR [28] and its corresponding robustness benchmarks [19, 40].

Model		C-10 err.	C-10.1 err.	C-10-C mCE	C-100 err.	C-100-C mCE
<i>Convolutional Neural Networks</i>						
ResNet-18 [15]	w/o ours	4.6	11.2	15.1	20.5	37.9
	w/ ours	4.1 (-0.5)	10.1 (-1.1)	14.6 (-0.5)	20.3 (-0.2)	38.0 (+0.1)
ConvNeXtv2-F [49]	w/o ours	4.3	11.0	11.1	24.2	38.9
	w/ ours	3.1 (-1.2)	7.9 (-3.1)	10.0 (-1.1)	19.1 (-5.1)	33.2 (-5.7)
<i>Pure Vision Transformer</i>						
DeiT-Ti [46]	w/o ours	10.5	22.2	23.0	35.3	51.8
	w/ ours	4.0 (-6.5)	9.8 (-12.4)	12.0 (-11.0)	24.7 (-10.6)	39.4 (-12.4)
PVT-T [47]	w/o ours	6.9	14.8	17.8	29.7	48.0
	w/ ours	3.6 (-3.3)	9.7 (-5.1)	11.7 (-6.1)	21.1 (-8.6)	35.9 (-12.1)
<i>Hybrid Architectures</i>						
PVTv2-B0 [48]	w/o ours	5.0	11.1	14.2	23.1	41.2
	w/ ours	3.5 (-1.5)	8.8 (-2.3)	12.6 (-1.6)	19.5 (-3.6)	34.2 (-7.0)
ConViT-Ti [10]	w/o ours	5.6	13.2	14.0	25.5	40.6
	w/ ours	3.2 (-2.4)	7.8 (-5.4)	9.6 (-4.4)	18.2 (-7.3)	31.5 (-9.1)
<i>Robust Architectures</i>						
RVT-Ti [35]	w/o ours	2.9	8.0	9.0	18.1	31.1
	w/ ours	2.4 (-0.5)	5.8 (-2.2)	7.2 (-1.8)	15.3 (-2.8)	27.3 (-3.8)
FAN-T [57]	w/o ours	3.4	9.1	10.0	19.8	34.3
	w/ ours	2.8 (-0.6)	8.0 (-1.1)	9.1 (-0.9)	18.2 (-1.6)	31.7 (-2.6)

and FAN-T [57], which are specifically designed to improve robustness. All models are trained in the same way for 300 epochs on the respective datasets and, if specified, extended by 50k generated images. The results are listed in Table 2.

Starting with BreastMNIST, the smallest dataset in our analysis, we observe a significant improvement in terms of error on clean data as well as mCE on corrupted data. The pure ViTs benefit the most from the additional data, lowering the error up to -14.1 on clean data and -10.0 on corrupted data. The CNNs and other transformer architectures also benefit from the additional artificial data, with the robust architectures showing the smallest improvement of about -4 . A consistent improvement is also observed for the other two medical datasets, with few exceptions. Similar to BreastMNIST, the pure transformers demonstrate the highest gain in performance as a result of additional generated images. When examining the result on EuroSAT, it can be seen that generated data can lead to further improvement even for already low error rates.

4.5 Comparisons on CIFAR-10 and CIFAR-100

In our last experiment, we extend our investigations for natural benchmarks beyond Tiny ImageNet to the smaller CIFAR-10 and CIFAR-100 datasets. In addition, we utilize CIFAR-10.1, CIFAR-10-C, and CIFAR-100-C. The purpose of this evaluation is to show the versatility of our GenFormer approach and its positive impact on a variety of architectural models. For all experiments on CIFAR, we use the same architectures as in Sect. 4.4. All models are trained in the same way for 300 epochs on the respective dataset and, if specified, extended by 100,000 generated images. The results are listed in Table 3.

The evaluation results show that pure ViTs benefit the most from the additional data. The error of DeiT-Ti on CIFAR-100 is reduced by more than 30%

relatively compared to the error without additional training data. The mCE on the corrupted data is also reduced by about 24%. In the case of PVT-T, there is a relative improvement of almost 30% and 25%, respectively, on the corrupted data. Significant improvements are also seen in the case of CIFAR-10. The hybrid and robust architectures also show improvements. These are in the relative range of about 10 to 30%. It should be noted that ResNet-18 as a CNN is almost unaffected by the additional generated data. Neither the error, nor the robustness improves significantly. In contrast to ResNet, ConvNeXtV2-F, which adapts many of the transformer design decisions for CNNs, responds much better to the generated data. There is a relative improvement of about 20% over CIFAR-100 and 15% over CIFAR-100-C. Again, the experiment on CIFAR exhibits the potential of using additional generated data to improve both the accuracy and robustness of especially ViTs. It demonstrates that the additional generated data helps transformers to learn more local features (see Appendix, Sect. D). Since CNNs naturally inherit a local bias, the gain from additional data diminishes for these networks. Furthermore, our GenFormer approach shows that for small datasets we are able to close the gap between CNNs and ViTs. In addition to experiments on small models ($< 15\text{M}$ parameters), we present further results on CIFAR-100, demonstrating scalability with models of up to 90M parameters and achieving state-of-the-art accuracy and robustness by combining GenFormer with established methods (included in the Appendix, Sect. C).

4.6 Ablation Study

In the subsequent ablation studies, we aim to investigate the influence of both the data generation network and the quality of the generated data on the accuracy and robustness of transformer classifiers. In addition, we investigate the effects of a longer training duration compared to a larger dataset.

Comparison of Different Generative Models. To begin, our initial focus revolves around assessing the impact of the data generation network on both the accuracy and robustness of transformer networks. To achieve this, we conduct a comparative analysis involving state-of-the-art Generative Adversarial Networks (GANs) [6, 23, 25, 44] and pixel-diffusion models [24, 27, 51]. We employ CIFAR-10 and ensure fairness by utilizing the best performing weights provided by the respective method’s developers. Since we focus on low-resolution datasets (32×32 for CIFAR and 64×64 for Tiny ImageNet [29]), our investigation is limited to pixel-diffusion models. For this comparative analysis between the mentioned models, we employ DeiT-Ti as the classifier subjected to two different training approaches. In the first scenario, the network undergoes an initial pre-training phase of 200 epochs on 100,000 generated images from the generative models, followed by a fine-tuning of 100 epochs on the real data. In the second scenario, we combine real and generated data and perform a single training phase of 300 epochs. Throughout the training process, we follow the training strategy provided by Liu et al. [32]. The results are listed in Table 4.

Table 4. Comparison of Generative Models [6, 23–25, 27, 44, 51] for generating 100,000 images to train a ViT [46] on CIFAR-10 [28].

Generative Models	FID* ↓	Gen. + FT		Mix	
		err.	mCE	err.	mCE
Baseline (w/o gen. data)	NA	10.5	23.0	10.5	23.0
<i>Generative Adversarial Networks</i>					
StyleGAN-XL [44]	1.85	10.1	21.1	6.5	17.1
StyleGAN2-ADA [25]	2.42	8.2	19.7	6.4	15.4
ReACGAN [23]	3.87	8.1	19.3	6.0	15.5
StyleGAN2 [25]	6.96	7.8	19.3	6.0	15.1
BigGAN [6]	14.73	7.5	18.8	5.6	14.8
<i>Diffusion Models</i>					
EDM-G++ [27]	1.64	6.4	17.2	4.0	12.1
PFGM++ [51]	1.74	6.5	17.4	4.1	12.5
EDM [24]	1.79	6.8	17.5	4.0	12.0

* FIDs are from the original publications.

Our results show that simply comparing FID scores is not sufficient when generating additional data to train ViTs. These findings further support observations of an earlier study by Ravuri et al. [39]. In the case of GANs, there is no discernible correlation between FID scores and accuracy. In both training variants tested, StyleGAN-XL [44], despite having the best FID (1.85), leads to the highest error on clean data (10.1/6.5) and on corrupted data (21.1/17.1). Conversely, BigGAN [6], which has the highest FID, leads to the lowest error (7.5/5.6). Nevertheless, the use of generated data consistently outperforms the baseline in training, regardless of the quality of the generated data.

In contrast to GANs, we observe significantly lower variability between FID values of diffusion models and the achieved error rates of the transformer network on CIFAR-10. Notably, there are no discernible performance disparities among the diffusion models. It is noteworthy that all diffusion models consistently outperform the compared GANs, leading to significant improvements in the classification network. For instance, the error of DeiT-Ti on clean data is reduced from 10.5 to 4.0, and on corrupted data, the mCE drops from 23.0 to 12.0. Given the lack of significant differences between the diffusion models, we consider the results of mixing real data with generated data when choosing our generator model. Considering the results in Table 4, we choose EDM [24] due to its competitive generation results and consequently use it alongside the mixing strategy for all our experiments.

Iterations vs. Samples. To validate that the presented improvements are a result of the proposed GenFormer approach and not by a longer training duration, the impact of the amount of training iterations is investigated. Figure 5

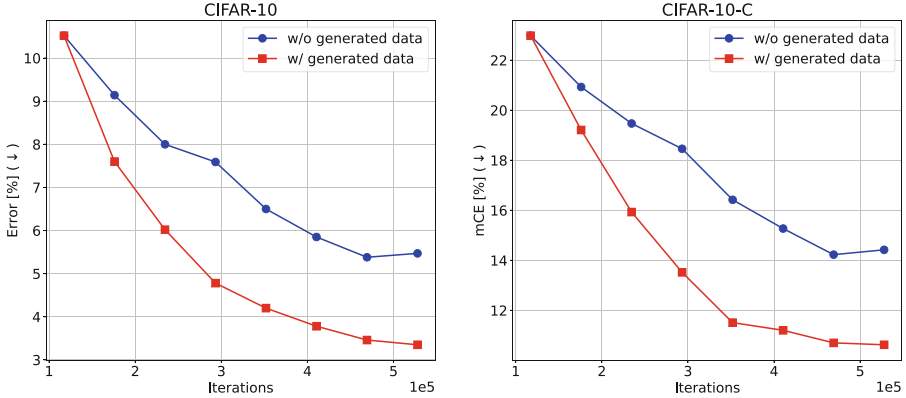


Fig. 5. Analysis of the impact of the duration of training (blue line) versus the number of data (red line) of DeiT-Ti [46] on CIFAR-10(-C) [19, 28]. (Color figure online)

shows the comparison of training runs with and without generated images of DeiT-Ti on CIFAR-10. The amount of iterations of the ViT training with generated images is controlled via the amount of additional generated images, whereas the training iterations for the training without generated images is controlled by the number of epochs. The graph starts at approximately $1.2 \cdot 10^5$ iterations, which corresponds to 300 epochs. For the baseline without generated data, the training time is gradually increased to 1,350 epochs (until beginning saturation) in steps of 150 epochs, which corresponds to a maximum of about $5.3 \cdot 10^5$ iterations. During this evolution, the error on clean data decreases from 10.5 to 5.5, and the mCE is reduced to 14.4. When using generated data, the size of the total data set is gradually increased from 50,000 (w/o generated images) to 225,000 samples (50,000 real with 175,000 generated images) in steps of 25,000 additional images. This stepwise increase leads to a reduction of the error to 3.3 and to a reduction of the mCE to 10.6 at 300 training epochs. In fact, training for more epochs slightly reduces the error and mCE. However, by comparing it to the results of the training with generated images, it is evident that significant improvements are induced by utilizing generated data.

5 Conclusion

We propose GenFormer, a generative data augmentation, in order to utilize the inherent robustness of ViTs in the small-scale data domain. We prove the effectiveness of data augmentation with generated images across various robustness and generalization benchmarks of small datasets including our newly introduced Tiny ImageNetV2, -R and -A test sets. By combining generative data augmentation with common data augmentation, knowledge distillation and architectural techniques, we show the straightforward applicability and synergistic potential of the proposed method. Furthermore, we showcase the consistent improvement

under various settings of heavy data limitations by training ViTs on small subsets of Tiny ImageNet. We conclude that GenFormer encourages a stronger focus on local features in early self-attention layers. Through the course of this work, we finally close the gap between ViTs and CNNs in terms of accuracy and robustness for limited-data scenarios.

Acknowledgements. This research was partly funded by Albert and Anneliese Konanz Foundation, the German Research Foundation under grant INST874/9-1 and the Federal Ministry of Education and Research Germany in the project M²Aind-DeepLearning (13FH8I08IA).

References

1. Al-Dhabyani, W., Gomaa, M., Khaled, H., Fahmy, A.: Dataset of breast ultrasound images. Data in brief (2020)
2. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. arXiv preprint 2304.08466 (2023)
3. Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., Veit, A.: Understanding robustness of transformers for image classification. In: International Conference on Computer Vision (ICCV) (2021)
4. Bilic, P., et al.: The liver tumor segmentation benchmark (lits). Medical Image Analysis (2023)
5. Bissoto, A., Valle, E., Avila, S.: GAN-based data augmentation and anonymization for skin-lesion analysis: a critical review. In: Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2021)
6. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (ICLR) (2019)
7. Chefer, H., Schwartz, I., Wolf, L.: Optimizing relevance maps of vision transformers improves robustness. In: Neural Information Processing Systems (NeurIPS) (2022)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
9. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2020)
10. d’Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: improving vision transformers with soft convolutional inductive biases. In: International Conference on Machine Learning (ICML) (2021)
11. Ebert, N., Reichardt, L., Stricker, D., Wasenmüller, O.: Light-weight vision transformer with parallel local and global self-attention. In: International Conference on Intelligent Transportation Systems (ITSC) (2023)
12. Ebert, N., Stricker, D., Wasenmüller, O.: Plg-vit: Vision transformer with parallel local and global self-attention. Sensors (2023)
13. Goyal, S., Rebuffi, S.A., Wiles, O., Stimberg, F., Calian, D.A., Mann, T.A.: Improving robustness using generated data. In: Neural Information Processing Systems (NeurIPS) (2021)
14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)

15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
16. He, R., et al.: Is synthetic data from generative models ready for image recognition? In: International Conference on Learning Representations (ICLR) (2023)
17. Helber, et al.: EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE JSTARS (2019)
18. Hendrycks, D., et al.: The many faces of robustness: a critical analysis of out-of-distribution generalization. In: International Conference on Computer Vision (ICCV) (2021)
19. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (ICLR) (2019)
20. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: a simple data processing method to improve robustness and uncertainty. In: International Conference on Learning Representations (ICLR) (2020)
21. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples (2021)
22. Jaipuria, N., et al.: Deflating dataset bias using synthetic data augmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
23. Kang, M., Shim, W., Cho, M., Park, J.: Rebooting ACGAN: auxiliary classifier GANs with stable training. In: Neural Information Processing Systems (NeurIPS) (2021)
24. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: Neural Information Processing Systems (NeurIPS) (2022)
25. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: Neural Information Processing Systems (NeurIPS) (2020)
26. Kermany, D.S., et al.: Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell (2018)
27. Kim, D., Kim, Y., Kang, W., Moon, I.C.: Refining generative process with discriminator guidance in score-based diffusion models. In: International Conference on Machine Learning (ICML) (2023)
28. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
29. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
30. Lee, S.H., Lee, S., Song, B.C.: Vision transformer for small-size datasets. arXiv preprint 2112.13492 (2021)
31. Li, K., Yu, R., Wang, Z., Yuan, L., Song, G., Chen, J.: Locality guidance for improving vision transformers on tiny datasets. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13684, pp. 110–127. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20053-3_7
32. Liu, Y., Sangineto, E., Bi, W., Sebe, N., Lepri, B., Nadai, M.: Efficient training of visual transformers with small datasets. In: Neural Information Processing Systems (NeurIPS) (2021)
33. Lovisotto, G., Finnie, N., Munoz, M., Mummadi, C.K., Metzen, J.H.: Give me your attention: Dot-product attention considered harmful for adversarial patch robustness. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)

34. Lu, Z., Xie, H., Liu, C., Zhang, Y.: Bridging the gap between vision transformers and convolutional neural networks on small datasets. In: *Neural Information Processing Systems (NeurIPS)* (2022)
35. Mao, X., et al.: Towards robust vision transformer. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
36. Paul, S., Chen, P.Y.: Vision transformers are robust learners. In: *AAAI Conference on Artificial Intelligence* (2022)
37. Pumarola, A., Sanchez-Riera, J., Choi, G., Sanfeliu, A., Moreno-Noguer, F.: 3dpeople: Modeling the geometry of dressed humans. In: *International Conference on Computer Vision (ICCV)* (2019)
38. Qin, Y., Zhang, C., Chen, T., Lakshminarayanan, B., Beutel, A., Wang, X.: Understanding and improving robustness of vision transformers through patch-based negative augmentation. In: *Neural Information Processing Systems (NeurIPS)* (2022)
39. Ravuri, S., Vinyals, O.: Classification accuracy score for conditional generative models. In: *Neural Information Processing Systems (NeurIPS)* (2019)
40. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: *International Conference on Machine Learning (ICML)* (2019)
41. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
42. Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S.N., Chellappa, R.: Learning from synthetic data: Addressing domain shift for semantic segmentation. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
43. Saryıldız, M.B., Alahari, K., Larlus, D., Kalantidis, Y.: Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2023)
44. Sauer, A., Schwarz, K., Geiger, A.: Stylegan-xl: Scaling stylegan to large diverse datasets. In: *ACM SIGGRAPH* (2022)
45. Tian, R., Wu, Z., Dai, Q., Hu, H., Jiang, Y.: Deeper insights into vits robustness towards common corruptions. *arXiv preprint 2204.12143* (2022)
46. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: *International Conference on Machine Learning (ICML)* (2021)
47. Wang, W., et al.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: *International Conference on Computer Vision (ICCV)* (2021)
48. Wang, W., et al.: Pvt v2: improved baselines with pyramid vision transformer. *Computational Visual Media* (2022)
49. Woo, S., et al.: Convnext v2: co-designing and scaling convnets with masked autoencoders. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2023)
50. Xie, Z., et al.: On data scaling in masked image modeling. In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2023)
51. Xu, Y., Liu, Z., Tian, Y., Tong, S., Tegmark, M., Jaakkola, T.: PFGM++: unlocking the potential of physics-inspired generative models. In: *International Conference on Machine Learning (ICML)* (2023)
52. Yang, J., et al.: MedMNIST v2-a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data* (2023)
53. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: regularization strategy to train strong classifiers with localizable features. In: *International Conference on Computer Vision (ICCV)* (2019)

- 54. Zhang, C., et al.: Delving deep into the generalization of vision transformers under distribution shifts. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
- 55. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: MIXUP: beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
- 56. Zhang, M., Levine, S., Finn, C.: MEMO: test time robustness via adaptation and augmentation. In: Neural Information Processing Systems (NeurIPS) (2022)
- 57. Zhou, D., et al.: Understanding the robustness in vision transformers. In: International Conference on Machine Learning (ICML) (2022)



Recognizing Patterns of Parkinson's Disease Using Online Trail Making Test and Response Dynamics – Preliminary Study

Artur Chudzik¹, Jerzy P. Nowacki¹, and Andrzej W. Przybyszewski^{1,2}(✉)

¹ Polish-Japanese Academy of Information Technology, Faculty of Computer Science,
86 Koszykowa Street, 02-008 Warsaw, Poland
przy@pja.edu.pl

² Department of Neurology, UMass Chan Medical School, 65 Lake Avenue,
Worcester, MA 01655, USA

Abstract. Neurodegenerative diseases (NDs), including Parkinson's (PD) and Alzheimer's (AD) disease are devastating conditions that affect millions worldwide, with the number of cases expected to rise significantly in the coming years. Despite considerable advancements in understanding their pathophysiology, etiology, and treatment, there is still a lack of effective disease-modifying interventions. Currently, no cure exists and there is an urgent need for modern tools that allow precise detection and objective severity scoring for the development of new therapeutic targets and approaches. Therefore, this study evaluates the effectiveness of an online version of the Trail Making Test Part A and B (TMT A and TMT B), incorporating time-based measures, to recognize cognitive and motor manifestations of Parkinson's disease severity. For validation, this research was conducted with 15 Parkinson's patients under care at UMass Chan Medical School. This study applied the TMT sensitivity to executive function impairments by measuring response and reaction times, to correlate these with stages of PD severity. Machine learning models (Naïve Bayes, Logistic Regression, Support Vector Machine, and Random Forest) were used to predict the disease severity based on TMT performance. Among these, Random Forest was the most effective, achieving scores with an Area Under the Curve (AUC) of 0.92 (80% accuracy), indicating good performance in distinguishing between mild and advanced stages of PD. Although limited by a small sample size, this preliminary study highlights the role of digital tools in enhancing PD diagnostics and monitoring. Future research with larger cohorts and longitudinal designs is essential to validate these preliminary findings and further develop digital diagnostics as crucial in the fight against neurodegenerative diseases.

Keywords: Machine Learning (ML) · Trail Making Test (TMT) · Diagnostic Accuracy · Parkinson's Disease (PD) · Time-Based Measurements · Web-Based Cognitive Testing · IRT · TTS

1 Introduction

Parkinson's Disease (PD) is a neurodegenerative condition that affects millions globally and it poses challenges in early detection and severity assessment. Traditional methods for detection and monitoring PD, despite their strengths, hold many limitations, such as invasiveness, high cost, or lack of accessibility [1, 2]. Recent advancements in digital biomarkers and machine learning present promising potential in bridging these gaps [3, 4].

However, research on online cognitive tests for Parkinson's disease (PD) has shown that current tools are underdeveloped and there is still an underexplored gap in using online cognitive tests and analyzing response dynamics for PD.

For example, Sousa (2022) emphasizes the lack of a common cognitive assessment battery for late-stage PD, and the need for tests that are quick, easy to use, and cover all relevant cognitive domains [5]. Here, authors recommend a cognitive assessment toolkit that considers the complex characteristics of PD, including being quick and easy to use, with minimized motor demands, and covering all relevant cognitive domains. Among the recommended instruments, the Trail Making Test corresponds well with these requirements [5].

The Trail Making Test Part A (TMT A) is a neuropsychological assessment tool primarily sensitive to processing speed and visual attention. It measures an individual's ability to rapidly connect a sequence of numbered circles in ascending order. TMT A focuses on basic scanning, attention, and motor speed.

The Trail Making Test Part B (TMT B) is a cognitive test known for its sensitivity to executive functions (involving attention, memory, visual search, motor function, and sequencing abilities). Trail B is generally more sensitive to executive functioning than TMT A since it requires multiple abilities to complete it.

But as Park et al. (2022) noted, the equivalency of paper-based and computerized tests cannot be assumed [6]. However, their findings support the comparability of TMT in computerized assessments and suggest this tool as a starting point for an early diagnostic tool. This conclusion is further supported by Mishra et al. (2022), who demonstrated that the digitized version of the TMT can determine cognitive-motor abilities and distinguish individuals with mild cognitive impairment and PD from healthy controls [7]. In their study, they observed correlations between TMT completion time and gait speed measured by a wearable accelerometer ($r = -0.4$, $p = 0.011$) and the Montreal Cognitive Assessment (MoCA) score ($r = -0.56$, $p < 0.01$).

Moreover, a study by Templeton et al. (2022) used fourteen tablet-based neurocognitive functional tests and machine learning model (decision trees) [7]. It allowed for the discrimination of PD from healthy controls (92.6% accuracy), and early and advanced stages of PD (73.7% accuracy). These results compare with current gold standard tools, such as standardized health questionnaires like the Unified Parkinson's Disease Rating Scale (UPDRS) with 78.3% accuracy and functional movement assessments with 70% accuracy.

Collectively, these findings underscore that a digital approach is feasible and allows for a comprehensive view of these conditions and their progression.

Therefore, this study explores the potential of the online version of Trail Making and machine learning models in identifying patterns indicative of Parkinson's Disease severity. By doing so, this research provides insights into quick, non-invasive, cost-effective, and accessible means of PD assessment and monitoring, potentially facilitating adapted interventions and earlier diagnosis.

2 Methods

We created an online version of Trail Making Test A and B [Fig. 1] [Fig. 2]. Then, we invited people with Parkinson's disease to solve this test. Our goal was to score their disease severity level (MILD or ADVANCED). The test took participants around two minutes to solve.

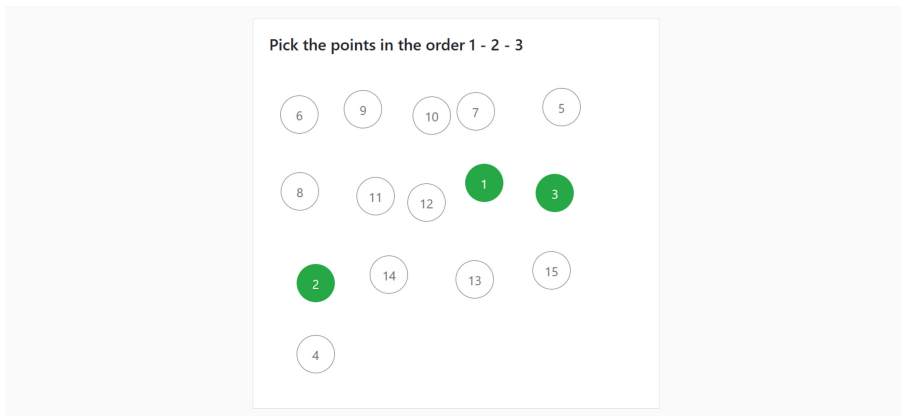


Fig. 1. The image presents an interactive cognitive task screen where the user is asked to pick points in a specific sequence. In TMT A, there are circles labeled with numbers.

In addition to the (a) number of mistakes, and (b) total time taken [TTS- time to submit, a standard measure in this test], we also recorded (c) instrumental reaction time [IRT- time to first selection] [8, 9]. That allowed us to assess the psychomotor speed and executive functions in two approaches.

In both approaches, reaction time was subtracted from the response time, to extract a pure psychometric component. In the first approach, we subtracted the time to complete TMT A from TMT B ($\text{TMT B} - \text{TMT A}$), that is a standard method in the clinical setup [10]. However, in the clinical context, the distinction between reaction and response times is not traditionally emphasized, and it's introduced as a novel method for this test.

Furthermore, in the second approach, we added the pure response times together ($\text{TMT A} + \text{TMT B}$) in order to evaluate machine learning models' performance using this variable and compare the results. In both approaches, longer times represented worse cognitive function.

All participants had confirmed Parkinson's disease diagnosis, and they were receiving treatment and advice from neurologists at UMass Chan Medical School. Eight participants had UPDRS III scores between 10 and 29 (indicative of mild symptoms of PD), and

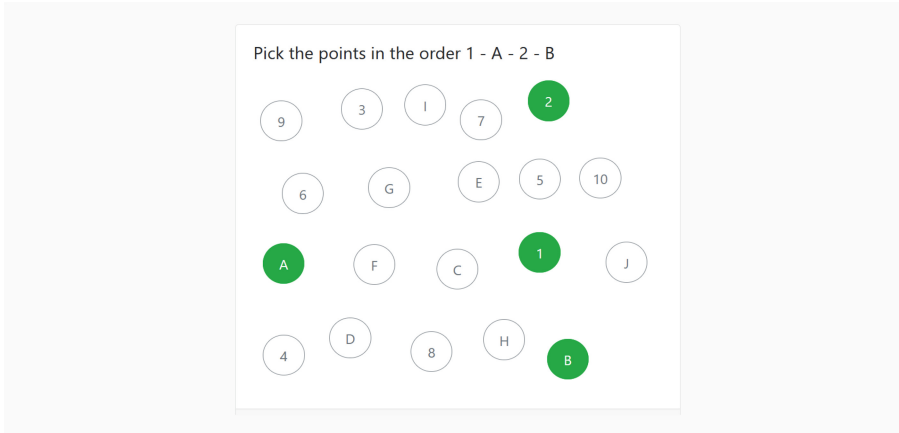


Fig. 2. The image presents an interactive cognitive task screen where the user is asked to pick points in a specific sequence. In TMT B, there are circles labeled with numbers together with letters.

seven participants with UPDRS III scores above 30 (indicative of advanced symptoms of PD). The sample size ($n = 15$) size makes available only preliminary comparisons and insights. However, we plan a study with a larger group to confirm these findings and explore other variables that may influence the results.

We analyzed aggregated data using IBM SPSS 29 software. We compared variables between patients with mild and advanced Parkinson’s disease, with p-value significant below 0.05.

Finally, we evaluated multiple ML models in a task of disease severity prediction. We implemented four machine learning models in Python using the library scikit-learn for modeling and metrics, together with pandas for data processing, and seaborn with matplotlib for visualizations [11–14].

The size of a dataset ($n = 15$) produced a risk of overfitting. To mitigate this risk, we considered simpler models rather than deep neural networks. Specifically, we selected Naïve Bayes, Logistic Regression, Support Vector Machine (SVM), and Random Forest. Moreover, before training, we extracted features importance to apply only the most relevant ones using Random Forest classifier.

Furthermore, we note that models’ performance can vary based on the specific characteristics of the dataset. Therefore, for a better overview of the performance, we analyzed and compared them alongside.

To address the consideration of hyperparameter tuning, we used GridSearchCV with cross-validation, and we limited the range and number of hyperparameters. We used Stratified K-Fold cross-validation method that ensured the same proportion of classes in each fold. The dataset was balanced. Models were trained on random 10 samples from the dataset and validated on 5 independent samples (train_test_split, allocating 1/3 of the data for testing before modeling). The test data was separated before applying k-fold CV.

The flowchart of applied methodology is presented in Fig. 3. This report presents outcomes, statistics, and ML models of this preliminary research.

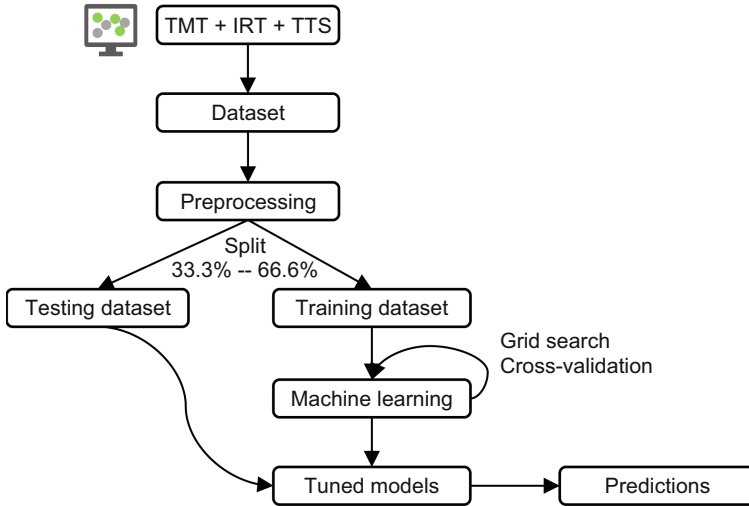


Fig. 3. Flowchart of applied methodology.

2.1 Features selection

In this study, the goal was to predict whether a participant belongs to MILD or ADVANCED group. However, with a small dataset, having too many features can lead to overfitting, where the model learns the noise in the training data instead of the actual signal.

Here, the introduction of a new composite variable can provide additional insights, especially when dealing with medical data sets and when optimizing for machine learning models. Therefore we experimented with two new composite variables (tmt_ba_medical_response_ms and tmt_ba_artificial_response_ms), that are designed to capture different aspects of TMT performance and narrow down the number of parameters.

Variable tmt_ba_medical_response_ms is calculated as the difference between the TMT B response time and reaction time, minus the difference between the TMT A response time and reaction time (Eq. 1).

$$\begin{aligned} \text{TMT_BA_MEDICAL_RESPONSE_MS} = & (\text{TMT_B_RESPONSE_MS} - \text{TMT_B_REACTION_MS}) \\ & - (\text{TMT_A_RESPONSE_MS} - \text{TMT_A_REACTION_MS}) \end{aligned} \quad (1)$$

Conceptually, it isolates the pure response component of the TMT B task from the TMT A task, attempting to adjust for basic reaction time to highlight more specific

cognitive processing or motor execution times involved in the more complex TMT B task compared to TMT A.

On the other hand, variable `tmt_ba_artificial_response_ms` sums the differences between the response and reaction times for both TMT B and TMT A (Eq. 2).

$$\begin{aligned} \text{TMT_BA_ARTIFICIAL_RESPONSE_MS} = \\ (\text{TMT_B_RESPONSE_MS} - \text{TMT_B_REACTION_MS}) \\ + (\text{TMT_A_RESPONSE_MS} - \text{TMT_A_REACTION_MS}) \end{aligned} \quad (2)$$

This approach combines the total time that is required to complete both tasks, potentially serving as a single measure of the cognitive and motor demands placed on the individual by both tests.

Both approaches allowed us to conduct modeling using clinically relevant parameters, such as error count for both tests, and singular time measurement in each experiment (either `tmt_ba_medical_response_ms` or `tmt_ba_artificial_response_ms`, accordingly).

It's important to note that in clinical settings, practitioners typically measure only response time in seconds using a pen-and-paper method for the TMT, without registering reaction time with high precision. This traditional approach does not capture the nuanced differences between reaction and response times that our composite variables do. Hence, our methodology offers a more accurate and insightful analysis of TMT performance, surpassing the conventional clinical setup's capabilities.

3 Results

The study involved fifteen participants. All of them had a confirmed Parkinson's disease diagnosis, and they were receiving treatment and advice from neurologists at UMass Chan Medical School. Eight participants had UPDRS III scores between 10 and 29 (indicative of mild symptoms of PD), and seven participants with UPDRS III scores above 30 (indicative of advanced symptoms of PD). Accordingly, patients were divided into two categories: MILD and ADVANCED.

3.1 Statistical Analysis

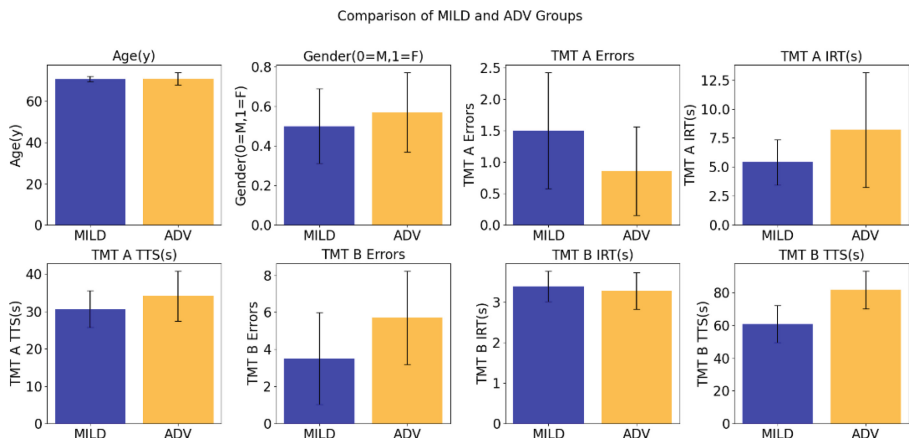
MILD had an average age of 70.75 years (Std. Error Mean [SE] = 1.306), with four females and four males. ADVANCED had an average age of 70.86 (SE = 3.074), with four females and three males [Table 1, Fig. 4].

Both groups have a similar age profile, with MILD at an average of 70.75 years and ADVANCED at 70.86 years. The p-value of 0.974 suggests there is no significant difference in age between the two groups. The gender distribution (represented as a proportion, with 0 for males and 1 for females) is slightly higher for females in the ADVANCED group (0.57) compared to the MILD group (0.50). However, the p-value of 0.800 shows this difference is not statistically significant.

On average, MILD patients made more errors (1.50) on the TMT A than ADVANCED patients (0.86). Despite this, the p-value of 0.599 indicates that the difference is not statistically significant.

Table 1. Comparison of the characteristics of patients in two groups – MILD and ADVANCED (ADV).

Variable	MILD (n = 8)	Std. Err.	ADV (n = 7)	Std. Err.	p-value
Age(y)	70.75	1.306	70.86	3.074	0.974
Gender(0 = M, 1 = F)	0.50	0.189	0.57	0.202	0.800
TMT A Errors	1.50	0.926	0.86	0.705	0.599
TMT A IRT(s)	5.408	1.956	8.199	4.951	0.591
TMT A TTS(s)	30.637	4.953	34.152	6.690	0.675
TMT B Errors	3.50	2.478	5.71	2.523	0.544
TMT B IRT(s)	3.399	0.377	3.282	0.452	0.844
TMT B TTS(s)	60.818	11.42	81.856	11.50	0.219

**Fig. 4.** Comparison of the characteristics of patients in two groups – MILD and ADVANCED (ADV).

The reaction and response times for TMT A are higher for the ADVANCED group compared to the MILD group, indicating slower performances. Specifically, reaction times average 5.408 s for MILD and 8.199 s for ADVANCED, while response times are 30.637 s for MILD and 34.152 s for ADVANCED. Neither difference is statistically significant, with p-values of 0.591 and 0.675, respectively. Interestingly, the reaction times in both groups were higher for TMT A than for TMT B, potentially due to the novelty aspect. This is because TMT A was presented as the first test, making users less familiar with it. As a result, they became more adjusted to TMT B, which may have shortened their reaction times.

Furthermore, ADVANCED patients tend to make more errors (5.71) on the TMT B than MILD patients (3.50), though this difference is not statistically significant (p-value = 0.544). Finally, for TMT B, the reaction and response times do not significantly

differ between groups, with MILD patients slightly faster in reaction times and slower in response times compared to ADVANCED. The p-values of 0.844 for reaction times and 0.219 for response times suggest these differences are not statistically significant.

3.2 Patterns in Variables

The connections between variables were explored using a heatmap form of Pearson correlation coefficients [Fig. 5]. This matrix presented a strong positive correlation of 0.69 between the age during the test and the reaction time in TMT A, suggesting that as age increases, the reaction time tends to increase as well.

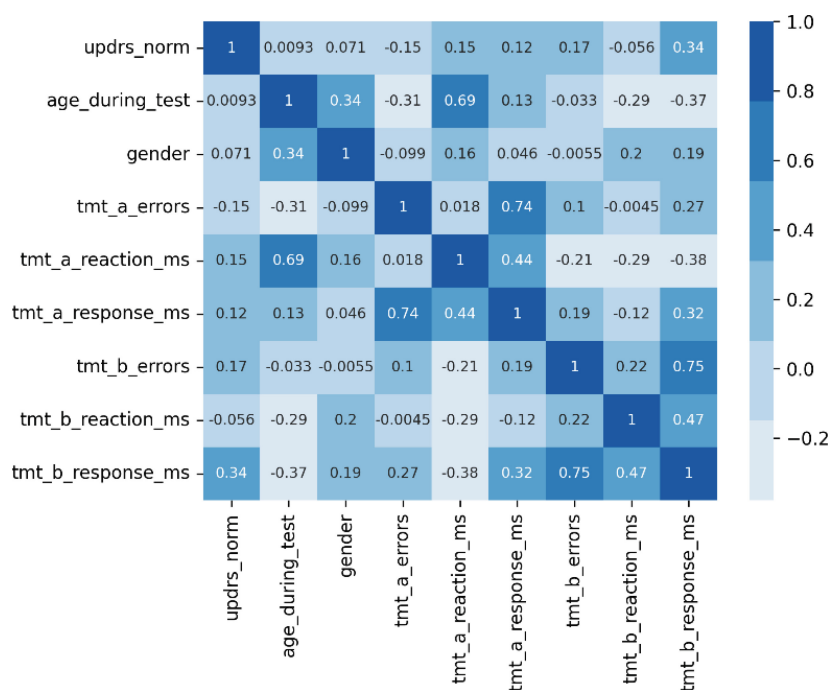


Fig. 5. A heatmap representing Pearson correlation coefficients. The darker the blue, the stronger the positive correlation; the closer to white or the presence of lighter blue indicates a weaker correlation. It's important to note that a correlation does not imply causation. These values simply indicate the strength and direction of the linear relationship between the pairs of variables.

Moreover, there is a very strong positive correlation of 0.74 between the number of errors made in TMT A and the TMT A response time, indicating that a higher number of errors is associated with a longer response time to complete TMT A.

Likewise, there is a very strong positive correlation of 0.75 between TMT B errors and TMT B response time, implying that as the number of errors increases, the total time to respond in TMT B also increases.

Additionally, there is a significant positive correlation of 0.47 between TMT B reaction time and TMT B response time, suggesting that longer reaction times are somewhat associated with longer total times to respond in TMT B.

Finally, a strong correlation exists between age during the test and TMT A reaction time, with a coefficient of 0.69, indicating that older participants may have slower reaction times in TMT A. The other correlations presented in the heatmap are weaker, meaning they show less of a linear relationship between variables. For instance, there's a weak negative correlation between TMT A Errors and Age during Test (-0.31), suggesting a slight tendency for older participants to make fewer errors on TMT A, although this relationship is not strong.

3.3 Machine Learning Models

To analyze how models learn, we used the Area Under the Curve (AUC). This is a measure of the overall performance of a classification model.

TMT Errors and Medical Response Dynamics

We incorporated the number of errors from TMT A and TMT B (`tmt_a_errors`, `tmt_b_errors`), together with medical response time (`tmt_ba_medical_response_time`) in order to predict UPDRS group (MILD or ADVANCED).

The Random Forest has the highest AUC of 0.92, indicating it has the best performance among the four classifiers in terms of ROC-AUC [Fig. 6]. This presents that it maximizes the true positive rate while minimizing the false positive rate better than the other classifiers in this set of ML algorithms.

Random Forest model performed noticeably better than the Logistic Regression, Support Vector Machine and Naïve Bayes. Random Forest correctly predicted 80% of the outcomes (accuracy: 0.8, precision: 0.75, sensitivity: 1.0, specificity: 0.5), which is significant when compared to other models [Table 2]. Random Forest predicted all positive cases correctly (1 true negatives, 3 true positives, and 1 false positive, with no false negatives).

The Logistic Regression and Support Vector Machine models both show a low accuracy of 0.4 (precision of 1.0 and sensitivity of 0.0), which means they were unable to correctly identify positive cases in this scenario. However, their specificity is at 1.0, indicating they could correctly identify all negative cases.

Furthermore, Naïve Bayes demonstrates the lowest accuracy among the compared models (accuracy: 0.2, precision: 0.33, sensitivity: 0.33, specificity: 0.0).

TMT Errors and Artificial Response Dynamics

In the second experiment, we incorporated the number of errors from TMT A and TMT B (`tmt_a_errors`, `tmt_b_errors`), together with artificial response time (`tmt_ba_artificial_response_time`) in order to predict UPDRS group (MILD or ADVANCED).

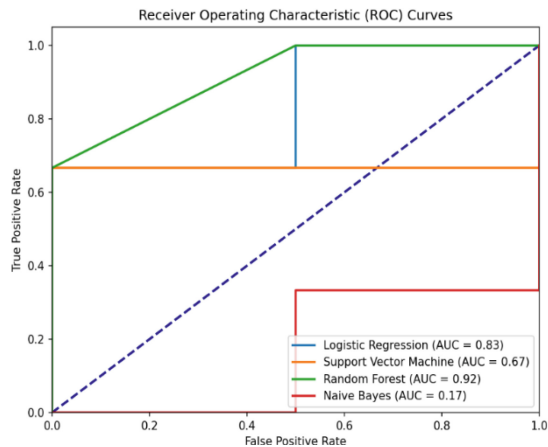


Fig. 6. The Area Under the Curve (AUC) is a measure of the overall performance of a classification model. Using Medical Response Time, Random Forest has the highest AUC of 0.92, indicating it has the best performance among the four classifiers. The Logistic Regression curve (blue line) is covered by the Support Version Machine (orange line) and Random Forest (green line) curve. The random change line represents the baseline performance of a random classifier (purple dashed line). (Color figure online)

Table 2. Comparison of models’ performance using tmt_a_errors, tmt_b_errors and tmt_ba_medical_response_ms. The table presents Accuracy (Acc.), Precision (Prec.), Sensitivity (Sens.), and Specificity (Spec.).

Model Name	Hyper-tuned Params	Acc.	Prec.	Sens.	Spec.
Logistic Regression	{ ‘C’: 1.0, ‘max_iter’: 100000, ‘penalty’: ‘l2’, ‘solver’: ‘lbfgs’ }	0.40	1.00	0.00	1.00
Support Vector Machine	{ ‘C’: 1.0, ‘kernel’: ‘linear’, ‘max_iter’: 100000 }	0.40	1.00	0.00	1.00
Random Forest	{ bootstrap’: True, ‘criterion’: ‘gini’, ‘max_features’: ‘sqrt’, ‘n_estimators’: 100 }	0.80	0.75	1.00	0.50
Naive Bayes	{ ‘priors’: None, ‘var_smoothing’: 1e-09 }	0.20	0.33	0.33	0.00

The Random Forest has the highest AUC of 0.92, achieving results comparable to the first experiment Fig. 7. After the analysis of feature importance in both models it seems that Random Forest scored tmt_ba_artificial_response_ms and tmt_ba_medical_response_ms similarly (0.65), placing tmt_b_errors on second (0.23), and tmt_a_errors on third place (0.11).

Repeatedly, Random Forest model performed better than the Logistic Regression, Support Vector Machine and Naïve Bayes. Random Forest correctly predicted 80% of the outcomes (accuracy: 0.8, precision: 0.75, sensitivity: 1.0, specificity: 0.5), which is significant when compared to other models Table 3. Random Forest predicted all positive cases correctly (1 true negatives, 3 true positives, and 1 false positive, with no false negatives).

The Logistic Regression and Support Vector Machine models both show a low accuracy of 0.4 (precision of 1.0 and sensitivity of 0.0), which means they were unable to correctly identify positive cases in this scenario. However, their specificity is at 1.0, indicating they could correctly identify all negative cases.

Furthermore, Naïve Bayes demonstrates the lowest accuracy among the compared models (accuracy: 0.2, precision: 0.33, sensitivity: 0.33, specificity: 0.0).

In conclusion, Random Forest stands out with the highest accuracy (0.8), good precision (0.75), perfect sensitivity (1.0), and a specificity of 0.5. Despite the lower specificity compared to Logistic Regression and Support Vector Machine, the high sensitivity and accuracy rates highlight its overall superior performance in predicting outcomes correctly.

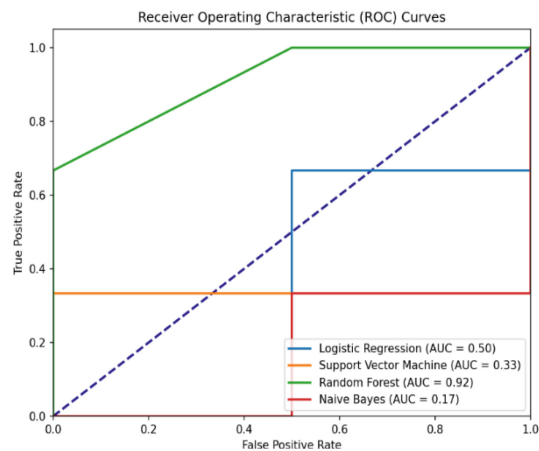


Fig. 7. The Area Under the Curve (AUC) is a measure of the overall performance of a classification model. Using Artificial Response Time, Random Forest has the highest AUC of 0.92, indicating it has the best performance among the four classifiers. The random change line represents the baseline performance of a random classifier (purple dashed line). (Color figure online)

Table 3. Comparison of models’ performance using `tmt_a_errors`, `tmt_b_errors` and `tmt_ba_artificial_response_ms`. The table presents Accuracy (Acc.), Precision (Prec.), Sensitivity (Sens.), and Specificity (Spec.).

Model Name	Hyper-tuned Params	Acc.	Prec.	Sens.	Spec.
Logistic Regression	{ ‘C’: 1.0, ‘max_iter’: 100000, ‘penalty’: ‘l2’, ‘solver’: ‘lbfgs’ }	0.40	1.00	0.00	1.00
Support Vector Machine	{ ‘C’: 1.0, ‘kernel’: ‘linear’, ‘max_iter’: 100000 }	0.40	1.00	0.00	1.00
Random Forest	{ ‘bootstrap’: True, ‘criterion’: ‘gini’, ‘max_features’: ‘sqrt’, ‘n_estimators’: 100 }	0.80	0.75	1.00	0.50
Naive Bayes	{ ‘priors’: None, ‘var_smoothing’: 1e-09 }	0.20	0.33	0.33	0.00

4 Discussion

This study preliminarily demonstrates the utility of the Trail Making Test with temporal measures in capturing the cognitive and motor impacts of bradykinesia in Parkinson’s Disease.

Importantly, this study evaluates both TMT A (processing speed and visual attention), and TMT B (cognitive functions) together with response dynamics (initial cognitive processing and decision-making speed). This is because in Parkinson’s disease, cognitive changes are independent from the motor symptoms development and thus have to be assessed independently [15]. This separation allows for the nuanced detection of PD’s impact. While TMT A focuses on motor speed and visual search abilities, requiring participants to connect numbered dots in sequence, TMT B adds a cognitive layer by alternating between numbers and letters. Incorporation of the reaction and response time measurements refines this approach, making this test more sensitive to slowed voluntary movement detection.

Therefore, there is the potential to approximate the real-time effects of dopamine through TMT performance. Impaired patterns observed in the test could be indicative of underlying disruptions in dopaminergic pathways, which are central to PD’s pathophysiology. This is particularly valuable given the challenge of directly assessing neurochemical changes in a clinical setting. By correlating TMT performance with known dopaminergic deficits, clinicians gain insights into the disease’s neurobiological patterns.

Interestingly, integration of insights from computational models of the brain and detailed studies on neuronal oscillations in PD patients provides a deeper understanding of the disease’s neural basis [16]. Mathiopoulos et al. presented that subthalamic beta oscillations are directly affected by both motor activity and therapeutic interventions such as dopamine replacement and deep brain stimulation (DBS) [17]. These beta oscillations, which are known to correlate with motor symptom severity, suggest a mechanism similar to the asynchronous process integration in the retina [16].

Just as the retina synchronizes processes to produce a coherent output from disparate sensory inputs, the subthalamic nucleus is crucial in coordinating motor commands disrupted by dopaminergic degeneration in PD. The TMT, by measuring reaction and response times, essentially assesses the efficiency of these neural synchronization patterns. Prolonged times might reflect the brain's struggle to integrate and synchronize neural processes efficiently, alike to the difficulties in processing and output synchronization.

To recognize these disrupted patterns, this study presents that Random Forest can be a helpful and accurate tool for this task. Random Forest are used because of their simplicity, ease of implementation, and their ability to perform well on a wide range of tasks with minimal hyperparameter tuning. Random Forest can capture interactions and nonlinear relationships between features, giving a possibility to model dopaminergic pathways disruptions through motor and cognitive data. It is worth noting that findings of this study align with other research that presents good performance of Random Forest in the disease severity classification task [9, 18, 19].

Here, it is important to note that this study has several limitations. In such a small group, even individual variability could account for these findings. These include the small size of the research groups, potentially limiting the generalizability. Therefore further, more detailed research (larger sample size, longitudinal study) is crucial for more detailed insights that help with the prevention of neurodegenerative diseases.

In light of this, the low performance of the Naïve Bayes model can be attributed to the small sample size of our dataset. Naïve Bayes relies on the assumption of a normal distribution of data and requires a sufficiently large dataset to accurately estimate the priors and likelihoods. Given the limited number of participants ($n = 15$), the data may not adequately capture the underlying distributions, leading to lower performance of the Naïve Bayes model. This issue might also relate to the fact that this dataset may not be representative of the broader population, further impacting the model's ability to generalize well.

Despite its limitations, this research shows the potential of a digital approach and contributes to the understanding and management of PD, particularly in customized interventions and early detection. TMT with temporal measures could be integrated into clinical practice or remote monitoring systems to better navigate and mitigate the impacts of PD.

Implementation of web version of TMT tests in clinical practice could offer a quick, non-invasive, and accessible method of assessing disease impact on cognitive and motor functions. Clinicians could use these tests for regular monitoring, enabling well-timed adjustments to treatment plans based on subtle changes in cognitive or motor performance.

In practice, implementing this approach requires creating a website that presents a TMT test with points labeled with letters and numbers, which participants must click in the correct order. This application needs to count the number of mistakes and record the start time of the first selection (IRT, e.g., `calling performance.now()`) and the time of the last selection (TTS, with the same method call). These metrics, along with TMT results, can provide valuable insights into delayed patterns in motor and cognitive responses.

Importantly, this technique can be enhanced by incorporating trajectory analysis of mouse movement. Moreover, there is an opportunity for further integration data going from wearable devices using sensors such as accelerometers and gyroscopes [7].

Integrating these assessments into telehealth platforms could facilitate remote monitoring, making it easier to track patient progress and intervene promptly. Such advancements could significantly enhance personalized care strategies, improving outcomes for PD patients.

To conclude, the results of this paper call upon the research community to explore these tools further and clinicians to consider their practical applications, given the significant diagnostic benefits that both they and their patients can gain.

Acknowledgments. The authors would like to thank all the participants who were involved in this study, including patients and doctors from UMass Chan Medical School.

Funding. This research received no external funding.

Institutional Review Board Statement

The study was reviewed and approved by the Institutional Review Board at UMass Chan Medical School (protocol code: IRB H0008962) to ensure compliance with the Declaration of Helsinki.

Informed Consent Statement

Informed consent was obtained from all subjects involved in the study.

Data Availability Statement. The data presented in this study are available on request from the corresponding author due to privacy and legal reasons.

Conflicts of Interest. The authors declare no conflicts of interest.




References

1. Talitckii, A., et al.: Comparative study of wearable sensors, video, and handwriting to detect Parkinson's disease. *IEEE Trans Instrum Meas.* **71**, 1–10 (2022). <https://doi.org/10.1109/TIM.2022.3176898>
2. Khare, S.K., Bajaj, V., Acharya, U.R.: PDCNNNet: an automatic framework for the detection of Parkinson's disease using EEG signals. *IEEE Sens J.* **21**, 17017–17024 (2021). <https://doi.org/10.1109/JSEN.2021.3080135>
3. Chudzik, A., Śledzianowski, A., Przybyszewski, A.W.: Machine learning and digital biomarkers can detect early stages of neurodegenerative diseases. *Sensors.* **24**, 1572 (2024). <https://doi.org/10.3390/s24051572>
4. Przybyszewski, A.W., Śledzianowski, A., Chudzik, A., Szlufik, S., Kozirowski, D.: Machine learning and eye movements give insights into neurodegenerative disease mechanisms. *Sensors.* **23**, 2145 (2023). <https://doi.org/10.3390/s23042145>
5. Severiano e Sousa, C., Alarcão, J., Pavão Martins, I., Ferreira, J.J.: Cognitive testing in late-stage Parkinson's disease: A critical appraisal of available instruments. *Appl. Neuropsychol. Adult.* **31**, 191–202 (2024). <https://doi.org/10.1080/23279095.2022.2114355>
6. Park, S.-Y., Schott, N.: The trail-making-test: Comparison between paper-and-pencil and computerized versions in young and healthy older adults. *Appl. Neuropsychol. Adult.* **29**, 1208–1220 (2022). <https://doi.org/10.1080/23279095.2020.1864374>

7. Mishra, R.K., Park, C., Zhou, H., Najafi, B., Thrasher, T.A.: Evaluation of motor and cognitive performance in people with parkinson's disease using instrumented trail-making test. *Gerontology*. **68**, 234–240 (2022). <https://doi.org/10.1159/000515940>
8. Chudzik, A., Drabik, A., Przybyszewski, A.W.: Investigating the Impact of Parkinson's Disease on Brain Computations: An Online Study of Healthy Controls and PD Patients. *Intelligent Information and Database Systems: 15th Asian Conference, ACIIDS 2023, Phuket, Thailand, July 24–26, 2023, Proceedings, Part II*. 235–246 (2023). https://doi.org/10.1007/978-981-99-5837-5_20.
9. Chudzik, A., Przybyszewski, A.W.: Classification of Parkinson's disease using machine learning with MoCA response dynamics. *Appl. Sci.* **14**, 2979 (2024). <https://doi.org/10.3390/app14072979>
10. Foki, T., Hitzl, D., Pirker, W., Novak, K., Pusswald, G., Lehrner, J.: Individual cognitive change after DBS-surgery in Parkinson's disease patients using Reliable Change Index Methodology. *Neuropsychiatrie*. **32** (2018). <https://doi.org/10.1007/s40211-018-0271-4>.
11. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*. **12** (2011)
12. Waskom, M.: Seaborn: statistical data visualization. *J Open Source Softw.* **6**, 3021 (2021). <https://doi.org/10.21105/joss.03021>.
13. Hunter, J.D.: Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **9**, 90–95 (2007). <https://doi.org/10.1109/MCSE.2007.55>
14. McKinney, W.: Data Structures for Statistical Computing in Python. 56–61 (2010). <https://doi.org/10.25080/Majora-92bf1922-00a>.
15. Przybyszewski, A.W., Nowacki, J.P., Drabik, A., Szlufik, S., Kozirowski, D.M.: IGrC: cognitive and motor changes during symptoms development in Parkinson's Disease Patients. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2020). https://doi.org/10.1007/978-3-030-42058-1_46.
16. Przybyszewski, A.W., Lindsay, P.S., Gaudiano, P., Wilson, C.M.: Basic difference between brain and computer: Integration of asynchronous processes implemented as hardware model of the retina. *IEEE Trans Neural Netw.* **18**, (2007). <https://doi.org/10.1109/TNN.2006.882814>.
17. Mathiopoulou, V., et al.: Modulation of subthalamic beta oscillations by movement, dopamine, and deep brain stimulation in Parkinson's disease. *NPJ. Parkinsons Dis.* **10**, 77 (2024). <https://doi.org/10.1038/s41531-024-00693-3>
18. Przybyszewski, A.W., Chudzik, A., Szlufik, S., Habela, P., Kozirowski, D.M.: Comparison of different data mining methods to determine disease progression in dissimilar groups of Parkinson's patients. *Fundam Inform.* **176**, 167–181 (2020). <https://doi.org/10.3233/FI-2020-1969>
19. Chudzik, A., Szymański, A., Nowacki, J.P., Przybyszewski, A.W.: Eye-Tracking and Machine Learning Significance in Parkinson's Disease Symptoms Prediction. In: Nguyen Ngoc Thanh and Jearanaitanakij, K. and S.A. and T.B. C.S. (ed.) *Intelligent Information and Database Systems*, pp. 537–547. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-42058-1_45.



Regularization of Interpolation Kernel Machines

Jiaqi Zhang^{} and Xiaoyi Jiang^{}^{}

Faculty of Mathematics and Computer Science, University of Münster,
Einsteinstrasse 62, 48149 Münster, Germany
`{j_zhan13,xjiang}@uni-muenster.de`

Abstract. Interpolation kernel machines belong to the class of interpolating classifiers that interpolate all the training data and thus have zero training error. Recent research shows that they do generalize well and have competitive performance. Several recent works proposed various ways of performance improvement for this decision model. In this work we investigate the generalization of interpolation kernel machines, which has not yet received enough attention. Our work is based on the popular regularization formulation with a penalty to the original loss function in order to constrain the model's capacity. We concretize six regularization methods. The experimental results clearly demonstrate the potential of generalization for classification performance improvement.

Keywords: Interpolation Kernel Machine · Regularization · Performance Boosting

1 Introduction

In machine learning, kernel-based techniques have a solid mathematical foundation and offer strong tools with applications in many domains. Apart from classification and regression [5, 14], they have also made significant contributions to other tasks like dimensionality reduction (e.g. PCA [11]), clustering [23], consensus learning [15], computer vision [12], and most recently, deep neural network research [8, 19].

In this work we study interpolation kernel machines. They are a class of interpolating classifiers with zero training error, i.e. they exactly fit the training set [2, 3, 7]. It is a common belief that such interpolating classifiers will result in overfitting. However, recent work shows compelling reasons to investigate these classifiers. For example, there is compelling evidence [24] that ensemble approaches function best when they are based on interpolating classifiers. One well-known instance is random forest. In order to comprehend the underpinnings of deep learning, Belkin [2] has recently emphasized the significance of interpolation (and its sibling over-parametrization). Interpolation kernel machines generalize effectively to unknown test data even if they have zero training error [3] (a behavior also often encountered in over-parametrized deep learning models).

Also from an application perspective there are good reasons to study interpolation kernel machines. They proved to be a viable alternative for deep neural networks, matching or even outperforming them in terms of performance while using less computational resources in training [7]. Recently, we have shown that interpolation kernel machines are very competitive to the widely used support vector machines [27]. In addition, interpolation kernel machines are also potentially beneficial for deep learning. Generally, while deep neural networks are powerful in feature learning, the de facto built-in neural network after feature learning is not necessarily the best choice for the classification or regression task. Thus, it is possible to replace this part by other decision models [20].

We have been working on further improving the performance of interpolation kernel machines. In [28] we proposed not to use their inherent multiclass classification capacity, but instead apply them for solving binary classification instances based on a multiclass-to-binary reduction. In [29] we studied multiple kernel learning, in particular the use of polynomial combined kernel functions. In [26] we presented a way of training set pruning, which turns out to boost the classification performance in addition to the increased efficiency in the test phase.

Regularization has been shown to be effective for neural networks [22]. In this work, we delve into the generalization capabilities of interpolation kernel machines, an area that has yet to receive adequate attention. Building upon established regularization formulations, we introduce penalty terms to the original objective function, thus effectively constraining the capacity of the model.

The remainder of the paper is organized as follows. We introduce the interpolation kernel machine in Sect. 2. Then, we present the regularization methods in Sect. 3. The experimental results follow in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Interpolation Kernel Machines

Here we introduce a technique to fully interpolate the training data using kernel functions, known as interpolation kernel machines [3, 7]. Note that this term has been often used in research papers (e.g. [6, 25]), where variants of support vector machines are effectively meant. For the sake of clarity we will use the term “interpolation kernel machine” throughout the paper.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\} \subset \mathcal{X}^m$ be a set of m training samples from some domain \mathcal{X} with their corresponding targets $\mathbf{Y} = \{y_1, y_2, \dots, y_m\} \subset \mathcal{T}^m$ in the target space, $(X, Y) \in \mathcal{X} \times \mathbb{R}$. The sets are sorted so that the corresponding training sample and target have the same index. Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a positive semidefinite kernel for some domain \mathcal{X} and \mathcal{F} be its associated RKHS (Reproducing Kernel Hilbert Space). A kernel-based learning method can be generally formulated by the following regularized empirical risk functional on \mathcal{F} :

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^m L(f(\mathbf{x}_i), y_i) + \lambda \mathcal{R}(f) \quad (1)$$

where L is the loss function of f . $\lambda \geq 0$ and \mathcal{R} is a regularization term. Since \mathcal{F} is possibly infinite dimensional, it is unclear if an efficient solution to this optimization problem can be found. Fortunately, a special form of regularization formulated in the following representer theorem reduces it to a finite dimensional optimization problem in the original space, which results in the interpolation kernel machine as a classification and regression tool.

2.1 Representer Theorem

Representer Theorem. Let $\Omega : [0, \infty) \rightarrow \mathbb{R}$ a strictly monotonically increasing function. We define $E : (\mathcal{X} \times \mathbb{R}^2)^m \rightarrow \mathbb{R} \cup \{\infty\}$ as an error function that calculates the loss L of f on the whole sample set with:

$$E(\mathbf{X}, \mathbf{Y}) = E((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)) = \frac{1}{m} \sum_{i=1}^m L(f(\mathbf{x}_i), y_i) + \Omega(\|f\|) \quad (2)$$

Then, any function $f^* = \arg \min_{f \in \mathcal{F}} \{E(\mathbf{X}, \mathbf{Y})\}$ admits a representation of the form:

$$f^*(\mathbf{z}) = \sum_{i=1}^m \alpha_i k(\mathbf{z}, \mathbf{x}_i) \quad \text{with } \alpha_i \in \mathbb{R} \quad (3)$$

The proof can be found in many textbooks, e.g. [5, 17].

While this statement of the representer theorem gives sufficient conditions on the regularizer, an interesting theoretical extension given in [1] shows that an interpolation problem (2) admits solutions representable in the form (3) *if and only if* the regularizer is a non-decreasing function of the Hilbert space norm, thus providing a complete characterization of regularizers that give rise to representer theorems. The recent work [16] studies the same problem in a more general context where the regularizer does not have to be norm-based.

2.2 Optimization Problem

Considering $\mathcal{X} = \mathbb{R}^d$ with the squared error loss, the kernel-based learning method as defined in (1) is concretized to:

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^m (f(\mathbf{x}_i) - y_i)^2 + \lambda \Omega(\|f\|) \quad (4)$$

by using the special form of regularization assumed in the representer theorem. By applying (3), we thus need to solve the equivalent problem:

$$\min_{\alpha \in \mathbb{R}^m} \sum_{i=1}^m \left(\sum_{j=1}^m \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - y_i \right)^2 + \lambda \Omega \left(\left\| \sum_{j=1}^m \alpha_j k(\cdot, \mathbf{x}_j) \right\| \right) \quad (5)$$

where $k(\cdot, \mathbf{x}_j) = \Phi(x_j)$ is the mapping of x_j to the feature space by the (typically not explicitly known) transformation Φ related to kernel k . $\alpha = (\alpha_1, \dots, \alpha_m)$

are the only learnable parameters, a real-valued vector with the same length as the number of training samples. Note that it is generally not easy to deal with the optimization problem in (5) since the second term is still formulated in the feature space and thus can only be explicitly computed for special cases of Ω like those based on L_2 norm. For this reason this term is usually ignored in the literature and instead the reduced optimization problem:

$$\min_{\alpha \in \mathbb{R}^m} \sum_{i=1}^m \left(\sum_{j=1}^m \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - y_i \right)^2 \quad (6)$$

is studied, thus *without any regularization* [3, 7, 26, 28].

2.3 Decision Model

The optimization in (6) can be easily done by solving the system of linear equations:

$$\mathbf{G}_m(\alpha_1, \dots, \alpha_m)^T = (y_1, \dots, y_m)^T \quad (7)$$

where $\mathbf{G}_m \in \mathbb{R}^{m \times m}$ is the kernel (Gram) matrix with the ij -th element $g_{ij} = k(x_i, x_j)$, $i, j = 1, \dots, m$. In case of a positive definite kernel k , the Gram matrix \mathbf{G}_m is invertible. Therefore, we can find the optimal α to construct f^* by:

$$(\alpha_1, \dots, \alpha_m)^T = \mathbf{G}_m^{-1}(y_1, \dots, y_m)^T \quad (8)$$

After learning, the interpolation kernel machine then uses the interpolating function from (3) to make prediction for test samples. Note that the learned function $f^* : \mathbf{X} \rightarrow \mathcal{T}$ is an interpolating one and satisfies:

$$f(\mathbf{x}_i) = y_i, \quad \forall i \in 1, \dots, m \quad (9)$$

In this work we focus on classification problems. In this case $f(z)$ is encoded as a one-hot vector $f(z) = (f_1(z), \dots, f_c(z))$ with $c \in \mathbb{N}$ being the number of output classes. This requires c times repeating the learning process above, one for each component of the one-hot vector. This computation can be formulated as follows. Let $\mathbf{A}_l = (\alpha_{l1}, \dots, \alpha_{lm})$ be the parameters to be learned and $\mathbf{Y}_l = (y_{l1}, \dots, y_{lm})$ target values for each component $l = 1, \dots, c$. The learning of interpolation kernel machine becomes:

$$\mathbf{G}(\underbrace{\mathbf{A}_1^T, \dots, \mathbf{A}_c^T}_{\mathbf{A}}) = (\underbrace{\mathbf{Y}_1^T, \dots, \mathbf{Y}_c^T}_{\mathbf{Y}}) \quad (10)$$

with the unique solution:

$$\mathbf{A} = \mathbf{G}^{-1} \cdot \mathbf{Y} \quad (11)$$

with a total of mc parameters to be learned. This is the extended version of (8) for c classes and results in zero error on training data. When predicting a

test sample z , the output vector $f(z)$ is not a probability vector in general. The class which gets the highest output value is considered as the predicted class. If needed, the output vector (z) can also be converted into a probability vector by applying the softmax function. Later, we will consider this option to enable the use of cross entropy loss function.

3 Regularization

Finding a model based on training data that can be well applied to unseen data is the main goal of machine learning. Regularization is one of the key techniques for achieving this goal [13, 21]. Technically, it can be realized in different ways, including arrangements made in terms of model architecture, learning process, and inference. In this work we resort to the widely used regularization formulation, which adds a penalty function to the original loss function in order to constrain the model's capacity.

3.1 Regularization Methods

We investigate the potential of regularization for interpolation kernel machines. In the general form it can be formulated as:

$$\min_{\alpha \in \mathbb{R}^{c \times m}} \sum_{i=1}^m L \left(f(\mathbf{x}_i) = \sum_{j=1}^m \alpha_j k(\mathbf{x}_i, \mathbf{x}_j), \mathbf{y}_i \right) + \lambda \mathcal{R}(f) \quad (12)$$

where α_j is the parameter vector of size c corresponding to training sample \mathbf{x}_j for all c classes and \mathbf{y}_j is the related ground truth labeling (one-hot vector). The optimization problem in (6) is a special form with quadratic loss function (and without regularization).

In this work we use both square and cross entropy loss function for (12). The use of cross entropy is justified due to its suitability for classification in general, although the derivation of interpolation kernel machine is based on square loss function. Indeed, the cross entropy variant turns out to achieve better performance in our experiments.

In total six regularization methods are studied:

- Weight decay regularization [13]: This is related to the ridge regression.

$$\mathcal{R}(f) = \sum_{i=1}^m \|\alpha_i\|^2$$

- Double backpropagation regularization [4]: Penalty of the squared L_2 norm of the gradient of the original loss term with respect to the inputs.

$$\mathcal{R}(f) = \left\| \frac{\partial}{\partial \mathbf{x}} L(f(\mathbf{x}), y) \right\|^2$$

- Jacobian regularizer: Penalty of the squared Frobenius norm of the Jacobian of the decision model output with respect to the input.

$$\mathcal{R}(f) = \|\mathbf{J}_f\|_F^2$$

There are two subvariants of this regularization: apply it to the decision model output directly (logits) [18] or the probability vector after softmax [9].

- Spectral regularization [22]: Use of a random projection to the Jacobian of the decision model output and then penalty of its squared L_2 norm.

$$\mathcal{R}(f) = \|P_{rnd}(\mathbf{J}_f)\|_F^2$$

where $P_{rnd}(\mathbf{J}_f) = \mathbf{J}_f^T \mathbf{r}$ and $\mathbf{r} \in \mathcal{N}(0, I_c)$ (matrix of size $c \times c$ with random numbers on the diagonal and zero otherwise). This method multiplies the elements of the Jacobian related to a particular class with a class-specific random number. Similar to Jacobian regularization, there are also two subvariants here: apply it to the decision model output without or with softmax (although only the first subvariant was considered in [22]).

Note that in all definitions above (except the weight decay regularization), the regularization is evaluated over all training samples and their sum, which is not explicitly shown for better conceptual clarity.

The regularization methods introduced above can be interpreted from different perspectives. Given a kernel k , the decision function $f^*(\mathbf{z})$ defined in (3) is a mapping from the original space \mathcal{X} to a m -dimensional feature space: $\mathcal{G}: \mathcal{X} \rightarrow \mathbb{R}^m$ by: $\mathcal{G}(\mathbf{z}) = (k(\mathbf{z}, \mathbf{x}_1), \dots, k(\mathbf{z}, \mathbf{x}_m))$. These features are then linearly combined based on parameters α_i . The weight decay regularization follows the standard definition that operates in this space. The rationale for the double backpropagation regularization is that if the input changes slightly the loss function should not change. One way of measuring this change is the derivative with respect to all the inputs. The four Jacobian-based regularization methods all aim to constrain the derivatives of the decision function along the different feature dimensions, which is geometrically helpful to reduce the probability of overfitting.

The introduction of regularization leads to optimization problems that have no analytic solution in contrast to the unregularized interpolation kernel machine (see Sect. 2.3). We use the solution of the unregularized variant (with minor random noise added) to initialize the iterative optimization.

3.2 Computation of Partial Derivatives

The computation of regularization terms requires a number of partial derivatives. In particular, the computation of the Jacobian matrix depends on the concrete kernel function. For the decision model output:

$$\textbf{Decision model 1: } f(\mathbf{x}) = \sum_{j=1}^m \alpha_j k(\mathbf{x}, \mathbf{x}_j) \quad (13)$$

Table 1. Overview of regularization methods.

Weight decay regularization (WDR)	
Double backpropagation regularization (DBR)	
Jacobian regularizer	logits (JRL)
	probabilities (softmax) (JRP)
Spectral regularizer	logits (SRL)
	probabilities (softmax) (SRP)

the elements of the Jacobian matrix are:

$$\frac{\partial f(\mathbf{x})}{\partial x_l}, \quad l = 1, \dots, d \quad (14)$$

When using a probability vector as decision model output:

$$\textbf{Decision model 2: } g(\mathbf{x}) = (p_1, \dots, p_c) = \text{softmax} \sum_{j=1}^m \alpha_j k(\mathbf{x}, \mathbf{x}_j) \quad (15)$$

the elements of the Jacobian matrix can be shown to become:

$$\frac{\partial g(\mathbf{x})}{\partial x_l} = p_l(1 - p_l) \frac{\partial f(\mathbf{x})}{\partial x_l}, \quad l = 1, \dots, d \quad (16)$$

We thus need the derivative (14) in order to compute the Jacobian matrix for both decision models. We exemplarily give it for two kernels (see Sect. 4): polynomial kernel (top) and Additive χ^2 kernel (bottom):

$$\frac{\partial f(\mathbf{x})}{\partial x_l} = \gamma d \sum_{j=1}^m \alpha_j (\gamma < \mathbf{x}, \mathbf{x}_j > + c)^{d-1} x_{jl} \quad (17)$$

$$\frac{\partial f(\mathbf{x})}{\partial x_l} = - \sum_{j=1}^m \alpha_j \frac{(x_l - x_{jl})(x_l + 3x_{jl})}{(x_l + x_{jl})^2} \quad (18)$$

where $l = 1, \dots, d$. The derivatives for other kernel functions can be determined similarly and are thus not presented here.

4 Experimental Results

4.1 Experimental Setting

We consider the following kernels for our experiments:

- Polynomial kernel: $k(\mathbf{x}, \mathbf{y}) = (\gamma < \mathbf{x}, \mathbf{y} > + c)^d$
- RBF kernel: $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$
- Laplacian kernel: $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|)$

- Additive χ^2 kernel: $k(\mathbf{x}, \mathbf{y}) = -\sum_{i=1}^d \frac{(x^i - y^i)^2}{x^i + y^i}$
- χ^2 kernel: $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma \sum_{i=1}^d \frac{(x^i - y^i)^2}{x^i + y^i}\right)$
- Sigmoid kernel: $k(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \langle \mathbf{x}, \mathbf{y} \rangle + c)$

where \mathbf{x} and \mathbf{y} are two samples with d features, x^i means the i th feature of sample \mathbf{x} , and analog y^i .

The regularization weight λ was determined to ensure a balanced order of magnitude of the two terms in Eq. (12): loss value $L(f)$ and regular value $\mathcal{R}(f)$.

Table 2. Description of UCI datasets.

dataset	# instance	# features	# classes
Acoustic	400	50	4
Australian	690	14	2
Autism_Adult	704	20	2
Biodeg	1052	41	2
Breast	286	9	2
Car	1728	21	4
Diagnosis	160	6	2
Flare-solar	1066	10	2
HCV	615	13	5
Ionosphere	351	34	2
Liver	345	6	2
Maternal	1014	6	3
New-thyroid	215	5	3
Raisin	900	7	2
Statlog	4435	36	6
Titanic	2201	3	2
Transfusion	748	4	2

Experiments were conducted on 17 UCI datasets (see Table 2 for an overview). These datasets are not of large size. In a recent work [26] it was shown that training set pruning is very helpful way to deal with large datasets. It reduces the efforts of training and in fact also testing in case of an instance-based classifier like interpolation kernel machine. In addition, the classification performance can even be boosted by using a small portion of the original training data only.

Table 3. Accuracy (%) of regularized interpolation kernel machine with square loss function.

dataset \ kernel	Polynomial							RBF						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
Acoustic	63.0	63.0	62.3	63.5	63.2	63.0	62.1	65.5	65.5	65.5	65.2	65.7	65.2	65.2
Australian	84.2	84.7	85.8	64.7	83.5	79.0	76.8	75.8	75.8	75.9	75.0	77.2	75.0	74.0
Autism_Adult	89.1	100.0	98.5	97.2	98.8	99.0	94.8	91.9	91.9	91.9	92.6	91.9	91.9	91.9
Biodeg	66.3	59.1	66.3	66.3	66.4	66.3	66.4	62.0	62.0	62.0	62.1	62.3	62.1	62.5
Breast	63.5	54.7	54.3	54.3	55.1	53.9	48.9	67.3	67.3	66.5	66.9	66.9	66.1	66.1
Car	70.0	70.0	70.6	70.7	70.8	70.5	70.5	83.3	73.1	82.9	83.2	83.3	82.5	83.5
Diagnosis	66.7	67.0	80.0	65.5	77.9	64.5	57.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Flare	70.9	70.2	44.0	70.1	70.9	77.9	70.9	36.2	36.2	35.0	35.4	35.4	36.2	36.2
HCV	89.3	87.6	91.0	80.7	89.3	81.9	89.3	91.9	91.9	91.7	90.2	93.4	90.3	93.9
Ionosphere	64.3	72.3	70.0	71.4	73.7	72.3	73.4	84.6	84.6	84.3	82.3	83.1	85.1	84.9
Liver	58.4	50.7	58.7	68.9	58.4	68.4	58.4	58.1	58.1	58.4	59.0	58.3	59.5	59.8
Maternal	41.3	44.1	60.8	54.1	53.4	65.9	46.9	29.1	58.1	61.0	49.9	53.5	49.4	53.3
New-thyroid	74.0	65.3	81.6	77.7	69.8	80.0	69.8	74.0	74.0	74.4	50.2	75.8	67.9	57.2
Raisin	76.3	74.9	72.0	74.8	65.6	54.4	77.4	51.7	56.7	73.2	63.6	62.8	56.2	60.7
Statlog	64.8	65.3	60.4	59.7	64.3	54.3	65.5	90.7	85.0	82.0	86.8	80.9	90.7	90.9
Titanic	63.0	75.8	52.3	73.2	73.5	75.8	73.5	62.6	47.4	41.2	52.6	40.7	44.2	47.7
Transfusion	60.8	63.9	56.9	53.0	73.9	53.0	73.9	54.4	72.6	68.8	72.9	72.5	68.6	64.9
average accuracy	68.6	68.7	68.6	68.6	71.1	69.4	69.2	69.3	70.6	71.5	69.9	70.8	70.1	70.2
superior percentage		52.9	58.8	52.9	64.7	52.9			41.2	58.8	41.2	70.6	58.8	70.6
dataset \ kernel	Laplacian							Additive χ^2						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
Acoustic	65.7	65.7	66.0	66.0	67.0	66.0	67.0	43.2	66.2	62.6	73.2	72.2	69.1	70.6
Australian	84.9	84.9	84.6	84.6	84.8	84.8	84.5	86.8	85.9	77.4	86.8	70.5	78.8	70.8
Autism_Adult	92.1	92.1	92.1	92.2	92.6	92.9	92.6	100.0	99.8	100.0	100.0	99.1	96.8	100.0
Biodeg	92.7	92.7	92.7	93.8	92.9	94.0	94.5	76.0	75.7	75.0	77.4	75.4	77.5	79.5
Breast	68.0	68.0	67.3	68.1	68.4	68.1	67.7	61.8	62.0	63.0	62.2	63.0	67.6	63.0
Car	83.3	83.2	87.5	86.5	84.7	86.3	87.5	77.4	70.0	68.5	70.5	65.7	65.7	71.4
Diagnosis	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	85.8	100.0	100.0	89.0	82.9	100.0
Flare	35.8	35.8	35.0	35.0	35.4	36.2	35.0	50.2	50.7	60.8	56.7	53.8	57.1	54.7
HCV	92.5	92.5	92.7	92.5	92.0	92.5	92.0	94.2	93.4	92.7	94.2	94.4	94.4	94.4
Ionosphere	86.0	86.0	86.0	85.7	85.7	86.0	86.3	86.6	85.0	86.0	86.6	89.3	89.1	85.1
Liver	61.0	61.0	61.6	58.1	58.1	62.2	61.0	73.9	68.1	61.8	73.9	71.6	73.2	74.2
Maternal	50.0	50.0	50.2	49.2	47.3	49.9	47.6	63.4	68.8	63.4	63.4	63.7	63.5	63.4
New-thyroid	91.6	91.6	91.2	91.6	91.2	90.2	86.5	88.4	88.2	88.4	88.4	85.3	89.3	88.40
Raisin	50.9	50.8	50.6	59.9	56.2	61.6	58.6	77.7	77.2	75.4	77.7	78.4	70.4	77.80
Statlog	90.2	90.2	89.1	89.1	89.9	90.3	90.2	85.4	86.9	79.4	84.3	78.8	78.8	78.9
Titanic	71.0	64.1	61.2	64.5	65.0	63.0	65.0	67.0	67.0	67.0	67.0	63.1	66.9	64.9
Transfusion	40.0	39.8	40.6	42.0	50.4	43.8	47.4	66.5	68.8	66.50	66.5	74.1	66.1	69.7
average accuracy	73.9	73.4	73.4	74.1	74.2	74.6	74.3	76.4	76.4	75.8	78.2	75.7	75.7	76.9
superior percentage		52.9	52.9	47.1	70.6	52.9			35.3	47.1	82.4	47.1	47.1	64.7
dataset \ kernel	χ^2							Sigmoid						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
Acoustic	64.1	64.1	64.1	64.4	64.4	63.9	64.4	70.3	73.0	81.6	80.0	83.2	83.0	82.7
Australian	81.1	81.1	81.1	81.1	81.0	80.7	80.4	77.0	83.2	85.5	81.7	83.5	84.5	84.2
Autism_Adult	92.2	92.2	92.2	92.7	92.4	92.2	92.4	98.5	100.0	92.4	97.5	96.8	97.5	97.1
Biodeg	73.2	73.1	72.5	73.1	73.1	73.1	73.2	66.3	65.7	66.4	66.8	66.1	66.4	66.4
Breast	66.9	66.9	66.5	66.5	67.7	66.4	65.7	69.7	65.6	64.2	62.1	69.7	60.0	69.7
Car	83.3	83.1	83.4	83.0	83.4	83.0	83.3	74.6	73.0	74.8	74.7	74.7	75.2	74.3
Diagnosis	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Flare	36.2	36.2	35.4	35.8	35.8	35.0	36.3	81.2	81.0	80.9	86.2	80.9	80.9	80.9
HCV	95.1	95.1	85.3	70.0	89.0	87.5	87.0	89.3	88.9	89.3	90.8	89.3	89.3	89.3
Ionosphere	85.1	85.1	85.7	85.7	85.4	84.3	84.6	70.4	71.7	74.3	73.7	74.3	74.6	74.6
Liver	56.0	56.3	56.9	56.9	55.4	56.0	55.4	58.0	56.3	58.4	59.8	58.4	59.8	58.4
Maternal	44.9	33.3	51.8	47.3	55.0	51.6	59.1	60.9	52.4	53.4	61.7	53.4	59.1	53.4
New-thyroid	89.3	89.3	89.3	89.8	87.4	88.9	89.3	69.8	87.9	89.8	90.7	91.8	93.0	90.0
Raisin	37.4	42.6	72.2	57.4	71.8	71.1	70.8	85.0	80.2	70.0	81.30	76.4	76.4	74.7
Statlog	91.3	91.3	91.3	85.5	86.4	92.5	91.3	75.3	74.0	80.3	75.3	69.3	74.2	76.2
Titanic	49.3	49.8	61.2	61.2	67.7	46.9	62.4	77.0	75.4	73.5	78.8	78.1	78.2	77.3
Transfusion	51.6	62.9	59.4	67.7	60.2	56.0	65.7	76.2	71.0	73.9	76.2	73.9	77.5	78.7
average accuracy	70.4	70.7	73.4	71.7	73.9	72.3	74.2	76.4	76.4	77.0	78.7	77.6	78.2	78.1
superior percentage		58.8	58.8	64.7	58.8	29.4	64.7		35.3	58.8	82.4	58.8	64.7	70.6

Table 4. Accuracy (%) of regularized interpolation kernel machine with cross entropy loss function.

dataset \ kernel	Polynomial							RBF						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
Acoustic	63.0	63.0	46.2	69.1	52.2	69.1	57.8	65.5	64.7	65.2	65.5	64.7	66.0	64.2
Australian	84.2	84.5	73.0	85.3	86.7	76.0	84.2	75.8	81.4	76.3	81.4	77.1	77.6	76.5
Autism_Adult	89.1	100.0	98.7	98.3	97.7	100.0	80.0	91.9	91.9	92.2	92.2	91.9	92.9	92.1
Biodeg	66.3	66.2	66.6	66.3	66.3	66.7	66.6	62.0	68.2	66.0	67.3	76.5	70.3	71.7
Breast	63.5	62.5	67.6	61.3	66.4	61.3	70.0	67.3	66.9	65.7	66.5	67.7	66.1	68.4
Car	70.0	69.6	77.1	71.7	77.1	70.0	70.1	83.3	82.4	80.3	93.5	94.7	94.7	94.5
Diagnosis	66.7	96.0	100.0	100.0	98.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Flare	70.9	51.7	46.1	57.9	51.8	51.6	63.7	36.2	36.6	35.4	35.4	34.9	35.8	34.9
HCV	89.3	83.6	91.2	89.3	83.3	82.8	89.3	91.9	91.5	77.7	90.7	79.5	84.0	73.5
Ionosphere	64.3	76.3	72.6	74.3	73.1	75.1	73.7	84.6	82.6	80.6	82.0	84.6	80.6	82.3
Liver	58.4	57.5	67.9	60.4	52.5	55.4	51.9	58.1	59.2	58.4	60.1	57.5	58.4	59.3
Maternal	41.3	42.4	44.9	64.7	54.6	44.7	49.0	29.1	36.4	60.0	31.3	57.4	57.8	61.0
New-thyroid	74.0	76.5	92.6	94.4	68.4	87.4	76.3	74.0	78.1	64.2	93.0	76.7	72.1	61.9
Raisin	76.3	73.2	62.9	71.1	69.0	84.8	68.9	51.7	41.4	60.1	52.1	65.8	65.0	65.3
Statlog	64.8	68.2	60.2	66.0	64.0	64.3	60.4	90.7	85.0	88.5	89.5	79.9	89.9	75.8
Titanic	63.0	61.8	59.4	78.0	67.7	60.6	60.4	62.6	61.2	50.4	63.4	59.8	50.2	49.7
Transfusion	60.8	49.3	51.5	72.4	70.3	71.3	54.7	54.4	55.2	65.9	55.4	67.7	70.0	74.3
average accuracy	68.6	69.6	69.3	75.3	70.6	71.8	69.2	69.3	69.6	69.8	72.3	72.7	72.4	70.9
superior percentage		47.1	58.8	82.4	58.8	58.8	52.9		52.9	47.1	70.6	64.7	58.8	58.8
dataset \ kernel	Laplacian							Additive χ^2						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
Acoustic	65.7	66.0	67.0	66.0	65.4	66.5	65.7	43.2	54.20	68.6	72.0	70.0	70.6	70.0
Australian	84.9	86.2	84.6	86.2	85.1	84.3	84.5	86.8	87.40	85.3	87.1	85.3	80.9	74.2
Autism_Adult	92.1	92.1	92.6	92.1	91.7	92.4	92.2	100.0	100	100.0	99.7	100.0	95.7	100.0
Biodeg	92.7	73.7	93.1	94.7	94.6	94.5	94.5	76.0	77.10	86.0	86.1	79.3	76.0	77.2
Breast	68.0	68.0	66.9	67.3	66.9	67.3	67.7	61.8	60.80	70.0	59.8	68.6	72.4	72.4
Car	83.3	73.2	80.3	86.7	84.9	87.6	97.6	77.4	77.4	77.50	77.4	78.9	78.6	79.1
Diagnosis	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100	100.0	100.0	100.0	100.0	100.0
Flare	35.8	35.8	36.6	35.4	35.4	35.0	35.0	50.2	51.6	56.3	53.0	50.6	47.2	62.7
HCV	92.5	91.0	92.0	91.5	92.0	92.5	92.5	94.2	84.9	92.7	94.2	93.8	89.8	89.3
Ionosphere	86.0	86.0	86.0	86.0	85.7	85.7	85.7	86.6	86.3	86.9	88.0	89.3	89.1	81.4
Liver	61.0	64.2	63.3	70.1	61.3	63.3	58.0	73.9	67.5	69.5	70.7	71.0	69.8	66.6
Maternal	50.0	61.9	53.0	60.2	60.4	49.3	48.3	63.4	67.0	59.1	67.4	66.9	66.4	64.5
New-thyroid	91.6	92.1	80.0	93.0	86.0	90.7	87.0	88.4	91.9	97.7	94.4	97.2	97.2	96.3
Raisin	50.9	66.2	53.1	53.2	56.6	55.7	51.7	77.7	78.6	77.3	79.8	76.4	76.1	78.0
Statlog	90.2	83.50	90.3	90.4	90.2	90.1	90.2	85.4	82.30	70.8	75.4	70.3	78.8	82.3
Titanic	71.0	61.2	55.4	78.2	70.7	75.6	70.9	67.0	75.9	67.0	71.5	77.6	70.0	77.6
Transfusion	40.0	55.6	57.2	64.4	62.2	53.6	64.0	66.5	66.3	72.0	72.7	73.7	72.5	73.3
average accuracy	73.9	73.9	73.6	77.4	75.8	75.5	75.6	76.4	77.0	78.6	79.4	79.3	78.3	79.1
superior percentage		70.6	64.7	82.4	52.9	52.9	41.2		64.7	64.7	76.5	70.6	58.8	70.6
dataset \ kernel	χ^2							Sigmoid						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
Acoustic	64.1	63.9	64.9	64.4	64.1	64.4	64.1	70.3	63.7	63.0	70.4	70.3	69.3	72.3
Australian	81.1	84.6	81.1	84.6	81.3	80.4	80.7	77.0	83.2	85.5	81.4	81.0	78.6	82.6
Autism_Adult	92.2	92.2	92.4	92.2	92.4	92.1	92.5	98.5	94.0	94.1	97.5	95.5	96.5	98.5
Biodeg	73.2	76.2	74.6	84.0	82.3	83.0	82.9	66.3	63.4	66.6	66.3	66.3	66.3	61.3
Breast	66.9	66.1	66.9	64.9	65.7	64.9	65.3	69.7	69.1	73.5	69.7	70.0	72.3	72.3
Car	83.3	72.0	83.3	93.2	93.8	93.8	93.6	74.6	76.0	75.4	74.6	76.6	74.0	75.6
Diagnosis	100.0	95.0	100.0	100.0	100.0	100.0	100.0	100.0	98.9	100.0	100.0	100.0	100.0	100.0
Flare	36.2	37.0	35.4	36.2	35.4	35.8	36.2	81.2	88.2	70.9	79.8	81.2	81.7	81.7
HCV	95.1	92.9	79.7	93.9	92.2	84.7	65.3	89.3	83.2	89.3	83.4	83.3	83.6	83.6
Ionosphere	85.1	84.6	84.0	84.6	83.7	80.9	86.0	70.4	74.0	74.3	81.4	81.4	81.4	81.4
Liver	56.0	57.8	57.2	60.7	56.6	56.0	58.4	58.0	56.3	58.4	58.9	60.0	59.8	57.8
Maternal	44.9	36.2	51.1	48.0	59.6	51.1	52.1	60.9	62.4	53.4	62.1	61.0	59.1	58.1
New-thyroid	89.3	93.0	86.0	96.3	85.1	85.6	85.1	69.8	87.9	89.8	95.8	91.2	93.0	92.3
Raisin	37.4	50.2	65.8	37.0	64.7	71.2	74.7	85.0	75.0	82.7	75.3	81.1	75.0	80.1
Statlog	91.3	88.4	88.3	90.6	76.4	90.8	88.4	75.3	76.4	68.1	74.7	74.2	74.2	70.7
Titanic	49.3	61.2	50.3	54.8	59.8	42.9	54.8	77.0	77.7	77.7	78.4	77.6	78.3	77.3
Transfusion	51.6	52.7	60.2	51.8	58.1	64.5	64.9	76.2	71.0	73.9	77.5	75.4	78.5	78.7
average accuracy	70.4	70.8	71.8	72.8	73.6	73.1	73.2	76.4	76.5	76.3	78.1	78.0	77.7	77.9
superior percentage		47.1	64.7	70.6	58.8	47.1	70.6		47.1	58.8	70.6	70.6	58.8	64.7

4.2 Experimental Results and Discussions

The experiments were conducted by 5-fold cross validation. The results (top-1 classification accuracy) are presented in Table 3 for the square loss function and Table 4 for the cross entropy loss function, respectively. In column “IKM” the performance of the unregularized version is shown for comparison purpose. In addition to the accuracy for each dataset, the average over all datasets is also given, with the best average accuracy marked bold. Note that there is no difference in performance if not regularized, i.e. for a given kernel function the column “IKM” in Table 3 and 4 are identical. That fact is due to the interpolating nature of this decision model with zero training error.

It is not surprising that the kernel functions differ in their performance. However, it is not the intention of this work to compare these kernel functions. Instead, we compare the behavior of the different regularization methods relative to each other and also relative to the unregularized version for a given kernel.

We focus on the average performance of all datasets for each method. Undoubtedly, regularization methods are useful for all kernel methods. In most cases, adding regularization terms is beneficial, but there are some exceptions, which cannot negate the merits of regularization terms. One example is Polynomial kernel with square loss, where DBR and JRL have the same accuracy as unregularized version. Another example is Additive χ^2 kernel with square loss, where the performance of WDR and DBR is disappointing. When coming to the details of each dataset, the results generally reveal the expected improvement by using regularization in almost all cases. Not all methods give expected results on all datasets for 6 kernels. Excepted examples are HCV data with RBF kernel for square loss and Breast data with Laplacian kernel for cross entropy kernel. In addition, the cross entropy loss function delivers superior results compared to the square loss function.

Since the cross entropy loss function is the favorable choice, our subsequent discussion will thus focus on this variant. In Fig. 1 we graphically show the average accuracy values for better clarity. While consistent improvement is observed for all kernel functions in average, it differs among the datasets for a particular kernel function. Thus, we also determined the percentage of datasets with performance improvement compared to the unregularized version, which is named “superior percentage” in Table 3, 4 and Fig. 1. It is computed within each column and defined by the percentage of each regularizer outperforming the unregularized version (the column “IKM”). Generally, there exists some positive correlation between the superior percentage and the classification accuracy, although not consistently.

To ease the comparison of the used regularization methods we plot the 36 (6 regularization methods \times 6 kernel functions) pairs (average performance, superior percentage) in Fig. 2. The higher the two values the better and the absolute optimum is the point (1, 1). Thus, looking at the point distribution of the regularization methods gives us a good impression of their relative performance. For instance, we observe four JRL points near the upper right corner that is highly desired while in contrast most SRL points are far away from the upper right cor-

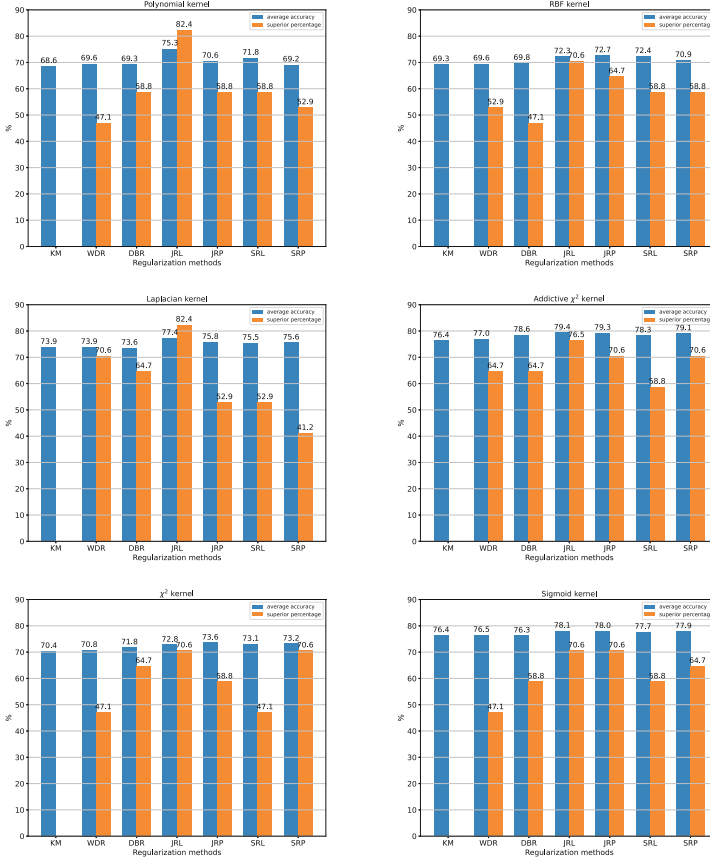


Fig. 1. Average accuracy and superior percentage.

ner. We can also perform a quantitative comparison by computing the average Euclidean distance of the 6 points from (1, 1) for each regularization method. In sorted order we obtain: JRL (0.346), JRP (0.450), SRP (0.480), DBR (0.484), SRL (0.509), WDR (0.528). Overall, we can thus conclude that JRL performs best. This may be explained by the clear rationale behind it. The two spectral regularization methods are variants of Jacobian regularizer with a stochastic component. They were introduced in [22], but not well motivated. The worst performance of WDR is not surprising. In the particular context of interpolation kernel machine there is no good reason to constrain the parameters.

We selected the 6 most imbalanced datasets according to imbalance ratio (IR, the ratio of the number of samples in the majority and minority classes) [10] and evaluated the performance by F_1 score. The sorted IR for the 6 datasets is: HCV(75.0), New-thyroid(5.0), Transfusion(2.8), Breast(2.6), Maternal(2.5), Flare(2.4).

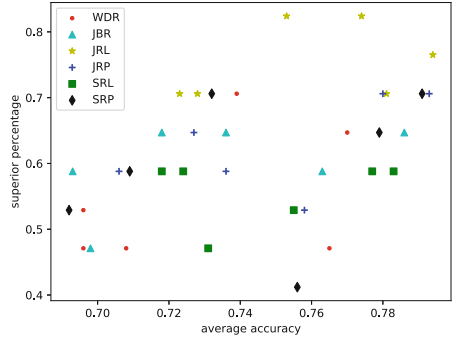


Fig. 2. Distribution of (average accuracy, superior percentage) pairs for relative performance comparison of regularization methods.

Table 5. F_1 score of regularized interpolation kernel machine with square loss function.

dataset \ kernel	Polynomial							RBF						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
HCV	0.257	0.409	0.253	0.242	0.242	0.257	0.250	0.502	0.369	0.403	0.476	0.476	0.479	0.479
New-thyroid	0.665	0.792	0.816	0.770	0.770	0.562	0.774	0.364	0.805	0.522	0.462	0.462	0.473	0.400
Transfusion	0.445	0.541	0.457	0.481	0.481	0.442	0.481	0.401	0.561	0.482	0.517	0.517	0.489	0.491
Breast	0.482	0.675	0.558	0.519	0.519	0.536	0.516	0.565	0.635	0.611	0.613	0.613	0.632	0.627
Maternal	0.530	0.584	0.582	0.577	0.577	0.444	0.568	0.273	0.544	0.319	0.343	0.343	0.351	0.311
Flare	0.462	0.593	0.616	0.563	0.563	0.607	0.574	0.281	0.537	0.473	0.518	0.518	0.515	0.518
average	0.473	0.599	0.547	0.525	0.525	0.475	0.527	0.398	0.575	0.468	0.488	0.488	0.490	0.471
dataset \ kernel	Laplacian							Additive χ^2						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
HCV	0.404	0.557	0.398	0.479	0.479	0.479	0.479	0.557	0.565	0.574	0.578	0.578	0.536	0.536
New-thyroid	0.883	0.938	0.979	0.965	0.965	0.965	0.965	0.784	0.922	0.829	0.824	0.824	0.824	0.842
Transfusion	0.327	0.620	0.588	0.591	0.591	0.595	0.590	0.467	0.582	0.548	0.577	0.577	0.577	0.563
Breast	0.582	0.688	0.587	0.640	0.640	0.636	0.640	0.529	0.647	0.645	0.624	0.624	0.669	0.669
Maternal	0.441	0.677	0.631	0.650	0.650	0.650	0.650	0.488	0.618	0.563	0.567	0.567	0.545	0.559
Flare	0.275	0.573	0.494	0.515	0.515	0.515	0.515	0.367	0.579	0.579	0.595	0.599	0.560	0.559
average	0.485	0.676	0.613	0.640	0.640	0.640	0.640	0.532	0.652	0.623	0.628	0.628	0.618	0.621
dataset \ kernel	χ^2							Sigmoid						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
HCV	0.592	0.628	0.646	0.684	0.684	0.684	0.684	0.189	0.325	0.189	0.211	0.189	0.189	0.189
New-thyroid	0.759	0.933	0.748	0.860	0.860	0.834	0.783	0.274	0.695	0.594	0.406	0.489	0.348	0.399
Transfusion	0.452	0.662	0.470	0.497	0.497	0.482	0.472	0.425	0.497	0.555	0.503	0.558	0.543	0.507
Breast	0.562	0.669	0.629	0.594	0.594	0.594	0.597	0.420	0.579	0.516	0.543	0.597	0.594	0.553
Maternal	0.336	0.636	0.355	0.349	0.349	0.371	0.323	0.232	0.486	0.391	0.432	0.349	0.444	0.413
Flare	0.281	0.534	0.523	0.476	0.476	0.470	0.470	0.415	0.633	0.491	0.613	0.616	0.564	0.599
average	0.497	0.677	0.562	0.576	0.576	0.572	0.555	0.326	0.536	0.456	0.451	0.466	0.447	0.443

Table5 and Table6 give the F_1 score for square loss function and cross entropy loss function, respectively. The higher values indicate more robust model.

Table 6. F_1 score of regularized interpolation kernel machine with cross entropy loss function.

kernel dataset	Polynomial							RBF						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
HCV	0.257	0.189	0.272	0.257	0.241	0.257	0.273	0.502	0.189	0.353	0.389	0.304	0.378	0.361
New-thyroid	0.665	0.628	0.547	0.664	0.716	0.562	0.659	0.364	0.556	0.687	0.764	0.670	0.637	0.677
Transfusion	0.445	0.425	0.428	0.437	0.440	0.442	0.446	0.401	0.469	0.522	0.518	0.526	0.501	0.511
Breast	0.482	0.630	0.441	0.567	0.592	0.536	0.482	0.565	0.602	0.562	0.614	0.605	0.624	0.601
Maternal	0.530	0.443	0.455	0.456	0.463	0.444	0.532	0.273	0.440	0.303	0.328	0.339	0.323	0.342
Flare	0.462	0.639	0.469	0.615	0.602	0.607	0.465	0.281	0.482	0.283	0.492	0.495	0.480	0.473
average	0.473	0.492	0.435	0.499	0.509	0.475	0.476	0.398	0.456	0.452	0.517	0.490	0.491	0.494
kernel dataset	Laplacian							Additive χ^2						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
HCV	0.404	0.189	0.392	0.366	0.449	0.430	0.430	0.557	0.225	0.189	0.576	0.076	0.377	0.220
New-thyroid	0.883	0.659	0.900	0.824	0.652	0.923	0.775	0.784	0.975	0.865	0.659	0.828	0.637	0.608
Transfusion	0.327	0.517	0.416	0.613	0.495	0.587	0.553	0.467	0.622	0.454	0.439	0.457	0.447	0.455
Breast	0.582	0.512	0.568	0.574	0.574	0.593	0.593	0.529	0.575	0.546	0.471	0.536	0.543	0.505
Maternal	0.441	0.534	0.517	0.630	0.626	0.608	0.616	0.488	0.543	0.448	0.520	0.321	0.442	0.474
Flare	0.275	0.535	0.267	0.513	0.505	0.504	0.496	0.367	0.611	0.360	0.440	0.431	0.473	0.494
average	0.485	0.491	0.510	0.587	0.550	0.608	0.577	0.532	0.592	0.477	0.517	0.441	0.487	0.459
kernel dataset	χ^2							Sigmoid						
	IKM	WDR	DBR	JRL	JRP	SRL	SRP	KM	WDR	DBR	JRL	JRP	SRL	SRP
HCV	0.592	0.383	0.559	0.656	0.638	0.648	0.646	0.189	0.172	0.189	0.211	0.179	0.223	0.223
New-thyroid	0.759	0.938	0.913	0.916	0.896	0.730	0.767	0.274	0.294	0.374	0.334	0.414	0.397	0.324
Transfusion	0.452	0.615	0.430	0.485	0.496	0.485	0.495	0.425	0.455	0.342	0.306	0.415	0.409	0.440
Breast	0.562	0.530	0.568	0.604	0.606	0.642	0.621	0.420	0.416	0.506	0.495	0.539	0.495	0.420
Maternal	0.336	0.564	0.288	0.379	0.398	0.365	0.361	0.232	0.391	0.405	0.513	0.503	0.560	0.399
Flare	0.281	0.534	0.289	0.542	0.541	0.544	0.533	0.415	0.491	0.597	0.603	0.498	0.504	0.600
average	0.497	0.594	0.508	0.597	0.596	0.569	0.571	0.326	0.370	0.402	0.410	0.425	0.431	0.401

Almost all results show an improvement after applying regularization terms. DBR techniques surpass the performance of other methods when using square loss function. And there is no outstanding regularizer for cross entropy loss function. The results for both loss functions indicate that regularized methods perform better than the unregularized IKM, even for imbalanced datasets.

5 Conclusion

Despite the interpolating nature, interpolation kernel machines prove to generalize well and have competitive performance. Several recent works proposed various ways of performance improvement for this decision model. In this work we investigated the generalization of interpolation kernel machines and demonstrated its potential for classification performance improvement.

The current work is limited to vectorial data only. In future we will extend it to non-vectorial domains, e.g. graphs with applications in chemistry and other

fields. In such domains, however, it is not trivial to define suitable realization of derivatives and Jacobian matrix computation. Then, the optimization problem in (5) becomes interesting. Despite some limitation (special cases of Ω like those based on L_2 norm) it enables a way of regularization in the feature space. It is a valuable contribution to working with structural data like graphs.

Acknowledgements. Jiaqi Zhang was supported by the China Scholarship Council (CSC). This research has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 778602 Ultracept.

References

1. Argyriou, A., Micchelli, C.A., Pontil, M.: When is there a representer theorem? vector versus matrix regularizers. *J. Mach. Learn. Res.* **10**, 2507–2529 (2009)
2. Belkin, M.: Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numer* **30**, 203–248 (2021)
3. Belkin, M., Ma, S., Mandal, S.: To understand deep learning we need to understand kernel learning. In: *Proc. of 35th ICML*, pp. 540–548 (2018)
4. Drucker, H., LeCun, Y.: Improving generalization performance using double back-propagation. *IEEE Trans. Neural Networks* **3**(6), 991–997 (1992)
5. Herbrich, R.: *Learning Kernel Classifiers: Theory and Algorithms*. The MIT Press (2022)
6. Houthuys, L., Suykens, J.A.K.: Tensor-based restricted kernel machines for multi-view classification. *Inf. Fusion* **68**, 54–66 (2021)
7. Hui, L., Ma, S., Belkin, M.: Kernel machines beat deep neural networks on mask-based single-channel speech enhancement. In: *Proc. of 20th INTERSPEECH*, pp. 2748–2752 (2019)
8. Jacot, A., Hongler, C., Gabriel, F.: Neural tangent kernel: Convergence and generalization in neural networks. In: *NeurIPS*, pp. 8580–8589 (2018)
9. Jakubovitz, D., Giryes, R.: Improving DNN robustness to adversarial attacks using jacobian regularization. In: *Proc. of ECCV, Part III*, pp. 525–541 (2018)
10. Khan, A.A., Chaudhari, O., Chandra, R.: A review of ensemble learning and data augmentation models for class imbalanced problems: combination, implementation and evaluation. *Expert Systems with Applications*, p. 122778 (2023)
11. Kim, C., Klabjan, D.: A simple and fast algorithm for L_1 -norm kernel PCA. *IEEE Trans. PAMI* **42**(8), 1842–1855 (2020)
12. Lampert, C.H.: Kernel methods in computer vision. *Found. Trends Comput. Graph. Vis.* **4**(3), 193–285 (2009)
13. Moradi, R., Berangi, R., Minaei, B.: A survey of regularization strategies for deep models. *Artif. Intell. Rev.* **53**(6), 3947–3986 (2020)
14. Motai, Y.: Kernel association for classification and prediction: a survey. *IEEE Trans. Neural Networks Learn. Syst.* **26**(2), 208–223 (2015)
15. Nienkötter, A., Jiang, X.: Kernel-based generalized median computation for consensus learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(5), 5872–5888 (2023)
16. Schlegel, K.: When is there a representer theorem? *Adv. Comput. Math.* **47**(4), 1–26 (2021). <https://doi.org/10.1007/s10444-021-09877-4>
17. Shalev-Shwartz, S., Ben-David, S.: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press (2014)

18. Sokolic, J., Giryès, R., Sapiro, G., Rodrigues, M.R.D.: Robust large margin deep neural networks. *IEEE Trans. Signal Process.* **65**(16), 4265–4280 (2017)
19. Tan, Y., Liu, H.: How does a kernel based on gradients of infinite-width neural networks come to be widely used: a review of the neural tangent kernel. *Int. J. Multimed. Inf. Retrieval* **13**(1), 8 (2024)
20. Thongsuwan, S., Jaiyen, S., Padcharoen, A., Agarwal, P.: ConvXGB: a new deep learning model for classification problems based on CNN and XGBoost. *Nucl. Eng. Technol.* **53**(2), 522–531 (2021)
21. Tian, Y., Zhang, Y.: A comprehensive survey on regularization strategies in machine learning. *Inf. Fusion* **80**, 146–166 (2022)
22. Varga, D., Csiszárík, A., Zombori, Z.: Gradient regularization improves accuracy of discriminative models. *Schedae Informaticae* **27**, 31–45 (2018)
23. Wang, R., Lu, J., Lu, Y., Nie, F., Li, X.: Discrete multiple kernel k-means. In: *Proc. of 30th IJCAI*, pp. 3111–3117 (2021)
24. Wyner, A.J., Olson, M., Bleich, J., Mease, D.: Explaining the success of AdaBoost and random forests as interpolating classifiers. *J. Mach. Learn. Res.* **18**, 48:1–48:33 (2017)
25. Xue, H., Chen, S.: Discriminality-driven regularization framework for indefinite kernel machine. *Neurocomputing* **133**, 209–221 (2014)
26. Zhang, J., Jiang, X.: Classification performance boosting for interpolation kernel machines by training set pruning using genetic algorithm. In: *Proc. of 13th Int. Conf. on Pattern Recognition Applications and Methods (ICPRAM)*, pp. 428–435 (2024)
27. Zhang, J., Liu, C., Jiang, X.: Interpolation kernel machine and indefinite kernel methods for graph classification. In: *Proc. of 3rd Int. Conf. on Pattern Recognition and Artificial Intelligence (ICPRAI)*. LNCS, vol. 13364, pp. 467–479 (2022)
28. Zhang, J., Liu, C., Jiang, X.: Interpolation kernel machines: Reducing multiclass to binary. In: *Proc. of 20th Int. Conf. on Computer Analysis of Images and Patterns (CAIP)*, Part I. *Lecture Notes in Computer Science*, vol. 14184, pp. 174–184. Springer (2023)
29. Zhang, J., Liu, C., Jiang, X.: Polynomial kernel learning for interpolation kernel machines with application to graph classification. *Pattern Recogn. Lett.* **186**, 7–13 (2024)



Task Success Classification with Final State of Future Prediction for Robot Control Planning

Taku Fujitomi^(✉) , Naoya Sogi , Takashi Shibata , and Makoto Terao

NEC Corporation, 1753, Shimonumabe, Nakahara, Kanagawa, Japan
{taku-fujitomi, naoya-sogi, m-terao}@nec.com, t.shibata@ieee.org

Abstract. We present a deep learning architecture designed to predict the success or failure of a robot control plan. The architecture accomplishes this by classifying the final state output of a future predictor using a recurrent neural network (RNN). In previous work, success or failure has been classified based on the output states of a recurrent state-space model (RSSM) and aggregated using the self-attention weights of each state. The success-or-failure predictor can serve as a constraint for enhancing robot control planning. However, it should be noted that the label of success or failure depends solely on the condition following the completion of the operation. As a result, classification should rely solely on the final output state of the future predictor. Yet, to focus on the final state in the classifier, it becomes necessary for the success-or-failure classifier to implicitly estimate the relative time from the state, since the state only represents a condition at a given time and lacks explicit time information. The challenge of time estimation can lead to poor performance. Considering this premise of success or failure, the proposed framework is designed to predict success or failure based solely on the final state of the RNN. This can be understood as a method that uses a fixed weight for aggregation over time steps. To improve performance with final state prediction, we propose the scheduling of classification loss weight. These changes are easily applicable to some reasonable RNN-based architectures. Our experiments on multi-object stacking tasks demonstrate that indeed effective for success-or-failure predictors.

Keywords: success-or-failure prediction · robot planning · recurrent neural network

1 Introduction

With dramatically progress in pattern recognition with deep learning architecture, the expectation of the use of pattern recognition manner for robot control have been growing significantly. Task and motion planning (TAMP) has been widely known as a practical approach for developing a control plan for executing long-term tasks such as the organization of multiple objects using a robotic

arm [9, 27, 31, 36, 39]. The key to the success of TAMP-based control plan execution is suitable design constraints aimed at preventing failure factors such as object grasp order, object collisions, object initial posture, and task termination conditions. In general, however, although designing these suitable constraints necessitates updating them based on the robot's behavior, monitoring the robot and its environment throughout the operation is challenging in actual situations. Furthermore, the constraints become increasingly complex as the task duration extends.

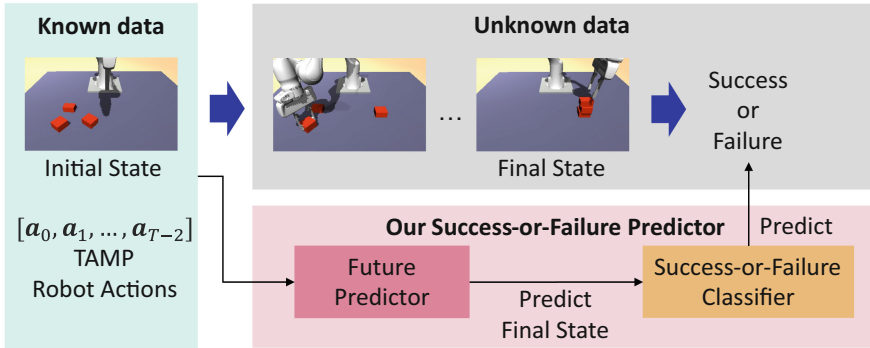


Fig. 1. Overview of the proposed framework. The proposed framework is a novel network architecture that can discriminate between success and failure based on the final state of RNN, as well as a scheme to adjust loss function weights for model training.

To address this challenge for task planning, the Future-predictive Identifier for Robot Planning (FIRP) [35] has been proposed as an effective framework to predict task success or failure classification as an outcome of executing particular plans prior to the enforcement of a control plan. The key of the existing framework is that the success-or-failure prediction results can be incorporated into control planning algorithms as supplemental key constraints, thereby improving the task success rate.

The FIRP framework employs a Recurrent State-Space Model (RSSM) [5, 11, 12, 14, 25, 26, 34] based on a Recurrent Neural Network (RNN) to predict future states at each time point. In the FIRP framework, self-attention module is used to estimate the weight for each state. Then, the weights of the calculated results are used to aggregate them. Finally, success or failure is classified [12, 35]. Since the success or failure of a plan candidate is generally determined based on the final state only, the classification module should rely exclusively on the output state of the final time from future prediction module. However, given the structure that estimates the weight of each state from the corresponding state, implicit time estimation from the state is required to base predictions on the final time. This implicit time estimation is a difficult task because the framework has not given any explicit annotations and/or training cues; there is a challenge that may have led to decreased accuracy.

In this paper, we propose a novel network architecture that discriminates between success and failure based on the final state of RNN, as well as a scheme to adjust loss function weights for model training (Fig. 1). The key is to explicitly incorporate the implicit assumption that “the final state is the critical cue” into the network architecture. By using only the final state of future predictions for classification, this proposed framework eliminates the need for weight estimation to concentrate on the final period. Our approach is built on top of our new base model that solves a domain gap problem in the classification part of FIRP. Experiments show that our framework is not only generic but also effective in improving accuracy for RNN-based success-or-failure predictors in multi-object stacking tasks dataset with MuJoCo simulated results [38]. The proposed manner is versatile as it is easily applied to sequential model derivatives. For LSTM-based networks, we further demonstrate that the proposed framework can significantly reduce training costs relative to conventional methods.

The contributions of this paper are as follows:

- We propose a new base model, an end-to-end network that solves a domain gap problem existing in FIRP.
- We propose a novel network architecture to discriminate success or failure from the final state of RNN and a scheme to adjust the weights of the loss function in model learning.
- We propose of a simple framework based on LSTM-based methods that can significantly reduced training cost.
- We demonstrate the effectiveness of the proposed framework for RNN-based success-or-failure prediction on MuJoCo simulated dataset.

2 Related Work

This section overviews robotic tasks using success-or-failure classification, and highlights our contributions.

2.1 Robotic Tasks Using Success-or-Failure Classification

The existing works for robotic tasks using success-or-failure classification can be roughly summarized into three categories; 1) environmental status classification, 2) action outcome classification, and 3) future predictive classification.

Environmental Status Classification. Environmental status classification is a task of identifying whether errors occur while observing the state of environment during robot operation [1, 2, 7, 10, 17, 18, 23, 28, 37, 45]. These methods are used to determine whether to carry out the recovery process. Typical information used as an input to a classifier is image or sound.

Action Outcome Classification. An action outcome classification is a task of estimating whether a given action will succeed in the current situation. There are two mainstream applications of an action outcome classification; 1) parameter estimation [21, 24, 42, 43], and 2) planning [4, 8, 20, 22, 29, 44].

For the parameter estimation, a classifier is used to estimate the parameters required to perform an action, such as a six degrees-of-freedom grasping pose.

For the planning, a classifier is used to determine the next actions to be inputted to a robot. A representative approach is to combine a classifier with the Cross-Entropy Method, a planning method [22, 32]. This approach is typically used for short-horizon tasks, such as object grasping. Recently, the combination of a classifier and a TAMP-based planner has been proposed for long-horizon tasks, such as object stacking and replacement [35].

Future Predictive Classification. Recently, Future-predictive Identifier for Robot Planning (FIRP) [35] has been proposed as an effective framework for success-or-failure identification for TAMP-based motion plans. The FIRP consists of a future prediction part using RSSM [11, 12, 26] and a classification part that predicts success or failure based on the prediction results.

In the training of the future prediction part of FIRP, two modules are trained in parallel: a module that predicts the prior distribution of states in the next step using the previous state and the planning sequence, and in addition to these, a module that predicts the posterior distribution by inputting images. They are optimized to predict the image features extracted by the image encoder from the images at each time as the correct states. The classification part integrates the actual image features at all times, weighted by self-attention [40], and outputs a binary score by all the combined layers (see Fig. 2(a)).

On the other hand, during inference, the posterior distribution prediction module using the initial state image is used at $t = 0$ to predict the state at each time from the initial image and the action sequence, and the prior distribution prediction module is used after the initial prediction. The classification part is populated with the predicted states at all times (see Fig. 2(b)).

The difference means that in FIRP, there is a domain gap problem between the inputs to the Classifier during training and inference.

2.2 Our Contributions

In contrast to the existing method [35], we propose a new base model to solve the domain gap problem in FIRP’s Classifier by aligning the input domains of Classifier during training and inference. Building on this model, we propose a new network architecture that discriminates success or failure based on the final state of the RNN and a scheme to adjust the weights of the loss function in model learning. By explicitly using only the final state of future predictions for classification, we eliminate the need for weight estimation focused on the final period. Furthermore, the proposed framework is a versatile framework and can be applied to LSTM-based architectures to reduce the training cost significantly compared to conventional methods.

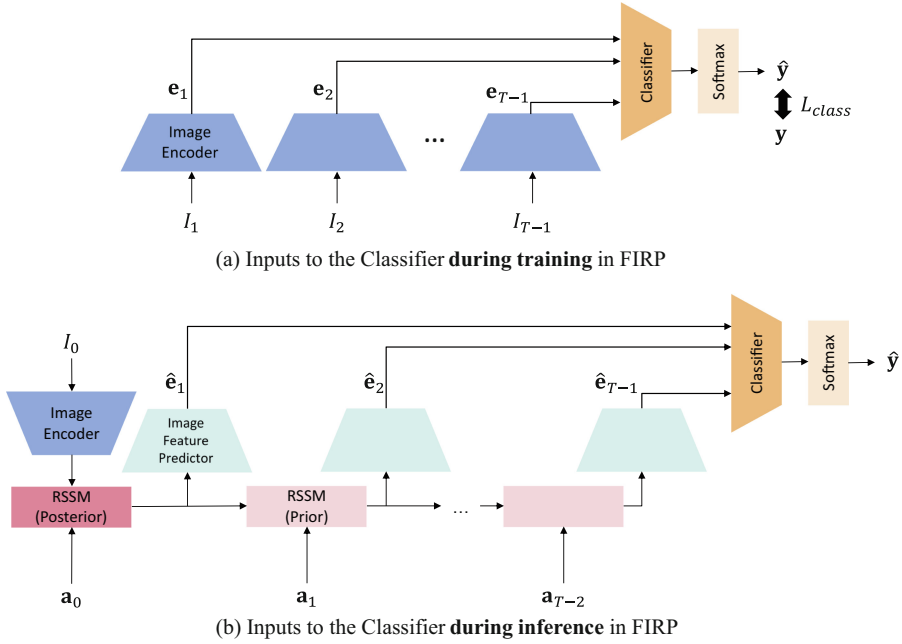


Fig. 2. The domain gap between the input to the Classifier during training and inference in FIRP [35]. (a) During training in FIRP, the inputs to the Classifier are image features \mathbf{e}_t . (b) Conversely, during inference, the inputs to the Classifier are image features $\hat{\mathbf{e}}_t$, which are predicted from the state of the RNN (using RSSM in FIRP). Thus, there exists a domain gap between the inputs to the Classifier during training and inference, depending on whether the input is based on actual observations or predicted values derived from I_0 and a sequence of actions \mathbf{a}_t .

3 Method

The goal of the proposed framework is to improve the success-or-failure classification accuracy while improving the computational cost for both training and inference by introducing the use of final state and an novel training protocols. In the following of this section, we first explain the overview of the proposed framework in Sect. 3.1. Next, we describe our key contribution, i.e. Success-or-failure predictor, and training protocols in Sect. 3.2 and Sect. 3.3, respectively.

3.1 Overview of Our Framework

We propose a simple-yet-effective framework for success-or-failure prediction based on the state vector that represents the RNN’s final state in light of the observation that the success or failure of a robotic task is determined solely by its final condition. Given that the success-or-failure classifier executes classification founded on the predicted state vector, it becomes particularly challenging

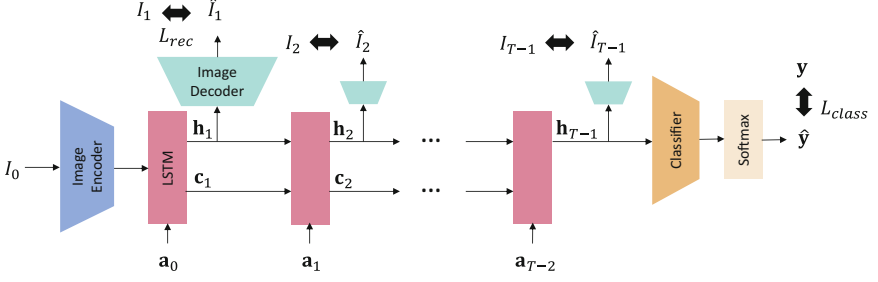


Fig. 3. The overview of our success-or-failure prediction framework with LSTM. Our network comprises an Image Encoder, a Recurrent Neural Network (LSTM or RSSM), an Image Decoder, and a Classifier. We adopt the same architecture during both training and inference.

to learn classification when the long-term prediction capability of the future prediction is low. Consequently, the proposed framework first undertakes learning of future prediction using the reconstruction loss between the ground truth image at each time and the image obtained through future prediction, and this begins prior to the learning phase with the success-or-failure classification loss.

As shown in Fig. 3, our success-or-failure prediction framework consists of an image encoder module, a recurrent neural network (LSTM or RSSM), an image decoder module, and a classifier module. In the following of this section, we will first describe each component of the proposed framework, and then explain the training algorithms for the proposed framework.

3.2 Success-Or-Failure Predictor

We describe the details of three key components of our success-or-failure predictor framework: i) Future predictor, ii) image decoder, and iii) success-or-failure classifier.

Future Predictor. We employ an RNN based on Long Short Term Memory (LSTM) [16, 33, 41] or RSSM [11–14] as a future predictor in our framework as shown in Fig. 3. Note that in Fig. 3, the proposed framework with LSTM is shown. We can apply the proposed framework to the RSSM-based networks by simply replacing the LSTM module with the RSSM module. When employing LSTM [16, 33], the state at each time can be written as:

$$\mathbf{h}_0 = f_{enc}(I_0), \quad (1)$$

$$\mathbf{c}_0 = \mathbf{0}, \quad (2)$$

$$(\mathbf{h}_t, \mathbf{c}_t) = g(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathbf{a}_{t-1}), \quad (3)$$

where the Image Encoder is denoted as $f_{enc}(\cdot)$, the RNN as $g(\cdot)$, and the initial state image as I_0 . \mathbf{a}_t is a planned robotic action at time t , and \mathbf{h}_t and \mathbf{c}_t are

the hidden state and cell state at t respectively. The initial state \mathbf{h}_0 employs a feature vector that is extracted from an initial image by the Image Encoder. Note that, in this paper, we use ResNet-50 [15] pre-trained on ImageNet [6] for image encoder.

When employing RSSM [12], the state at each time during inference is written as:

$$(\mathbf{h}_0, \mathbf{s}_0) = \mathbf{0}, \quad (4)$$

$$(\mathbf{h}_1, \mathbf{s}_1) = g_{pos}(f_{enc}(I_0), \mathbf{h}_0, \mathbf{s}_0, \mathbf{a}_0), \quad (5)$$

$$(\mathbf{h}_t, \mathbf{s}_t) = g_{pri}(\mathbf{h}_{t-1}, \mathbf{s}_{t-1}, \mathbf{a}_{t-1}), \quad (6)$$

where $g_{pos}(\cdot)$ represents a module predicting state posteriors using the image as an input, and $g_{pri}(\cdot)$ as the module predicting state priors without using the image as an input. \mathbf{s}_t is a stochastic state at t .

Image Decoder. In the training phase, our network reconstructs images which represent the target condition at each time.

$$\hat{I}_t = f_{dec}(\mathbf{h}_t), \quad (7)$$

where $f_{dec}(\cdot)$ is Image Decoder, and \hat{I}_t is an image at t . In this paper, the Image Decoder $f_{dec}(\cdot)$ is one fully connected layer and three transposed convolutions with rectified-linear-unit activation.

Success-Or-Failure Classifier. We employ the same classifier architecture as in FIRP [35]. In FIRP, the Classifier used inputs from all the states. In contrast, the proposed framework utilizes only the final state as the input. Formally, this alteration can be also interpreted as fixing the weight of each state to 1 in the final state and 0 for all other states. In this sense, the proposed method is natural extension of the existing method FIRP; We can inherit the strengths of FIRP. The network architecture of the Classifier is shown in Fig. 4. In the proposed framework, the self-attention part shown in the green section can be skipped because the resulting attention weight is always 1 when the number of input states over time dimension is 1.

To obtain the label probabilities $\hat{\mathbf{y}}$, we apply the Softmax function for the output logits of the Classifier:

$$\mathbf{logits} = f_{class}(\mathbf{h}_{T-1}), \quad (8)$$

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{logits}), \quad (9)$$

where $f_{class}(\cdot)$ is the Classifier, and T is the sequence length of a plan. In the classification, the class labels are success and failure, hence \mathbf{logits} and $\hat{\mathbf{y}}$ are both 2-dimensional vectors, and the classification result is chosen as the class with the maximum value in $\hat{\mathbf{y}}$.

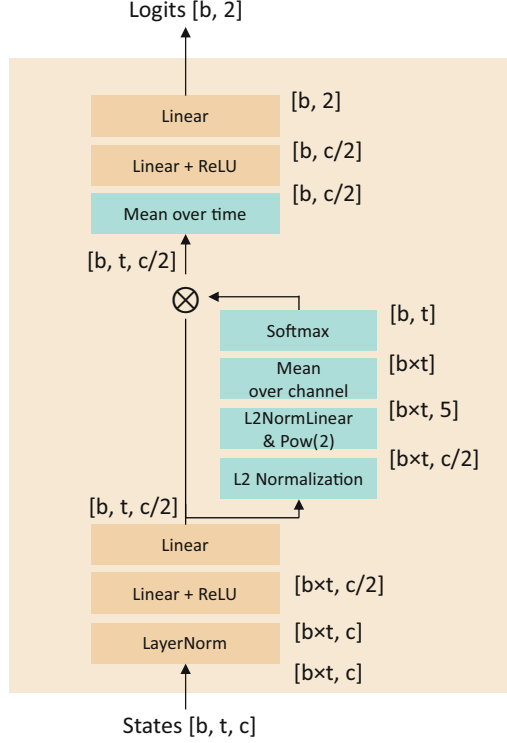


Fig. 4. The network architecture of the Classifier. In L2 Norm Linear, the learnable weight is normalized along the dimension corresponding to the input channels. The results after the L2 Normalization and the following L2 Norm Linear are equivalent to the cosine similarities between inputs and the learnable vectors. The b , t , and c represent the batch size, number of input states, and number of channels, respectively. In the proposed framework, since the input to the Classifier is only the final step, t is 1, and the green part, which calculates the weight of each time, is not necessarily used.

3.3 Training Protocols for Our Success-or-Failure Prediction Framework

We describe training protocols for our success-or-failure prediction framework. Specifically, we explain a loss function for training and classification weight scheduling during our training.

Loss Function. The loss function for the proposed framework is composed of two terms: 1) reconstruction loss and 2) classification loss. We calculate a reconstruction loss using the images capturing the condition around the robot at each time. We adopt the mean squared error (MSE) between the ground truth

images and the predicted images:

$$\mathcal{L}_{rec} = \frac{1}{T} \sum_{t=1}^T \sum_{w=1}^W \sum_{h=1}^H \sum_{c=1}^C (I_t(w, h, c) - \hat{I}_t(w, h, c))^2, \quad (10)$$

where W , H , and C represent the width, height, and number of channels of the image, respectively. Here, T is the sequence length of a plan.

For the classification loss, we use the cross-entropy loss. Given N as the number of classes, \hat{y}_i as the probability corresponding to the i -th class in the Classifier output $\hat{\mathbf{y}}$, and y_i as the ground truth binary label of the i -th class, the classification loss is defined as:

$$\mathcal{L}_{class} = - \sum_{i=1}^N y_i \log \hat{y}_i. \quad (11)$$

Since the proposed method is a versatile framework, it can be used for various architectures such as LSTM and RSSM. In the proposed framework with LSTM, the overall loss is defined as the summation of the reconstruction loss and weighted classification loss, which is simply given by

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{cls} \mathcal{L}_{class}, \quad (12)$$

where λ_{cls} is a weight constant for the classification loss.

For the proposed framework with RSSM, Kullback-Leibler divergence between $g_{pos}(\cdot)$ and $g_{pri}(\cdot)$ are additionally calculated in the same manner as in FIRP [35], which is formally given by

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{cls} \mathcal{L}_{class} + \mathcal{L}_{KL}. \quad (13)$$

In addition, as in FIRP [35], the proposed framework uses the overshooting loss [3, 12] for accurate long-horizon prediction. In latent overshooting manner, a stochastic features are computed from latent variables in RSSM framework. Details of the implementation are described in [3, 12].

Classification Weight Scheduling. To ensure that the final state approximates the prediction of the actual final condition at the start of the training with the classification loss \mathcal{L}_{class} , the proposed framework starts the training solely with \mathcal{L}_{rec} before using both \mathcal{L}_{class} and \mathcal{L}_{rec} . To this end, the λ in Eq. (12) is defined as.

$$\lambda_{cls} = \begin{cases} 0, & \text{if } e < s, \\ 1, & \text{otherwise,} \end{cases} \quad (14)$$

where e is the current epoch and the hyperparameter s is the epoch at which \mathcal{L}_{class} is activated.

4 Experiments

We demonstrate the effectiveness of the proposed framework based on success-or-failure prediction experiments for the stacking task of multiple objects. In the following of this section, we first describe the settings of our experiments in Sect. 4.1. Then, experimental results and discussions are described in Sect. 4.2.

4.1 Settings

In the following, we explain the dataset for our experiments, evaluation protocols, and implementation details.

Datasets. In order to compare the performance of the proposed framework with the existing method [35], we construct a dataset for stacking tasks, utilizing a Task and Motion Planning (TAMP) method [36]. Examples of images in our dataset for this experiment are shown in Fig. 5. We set up a task of stacking three cuboids placed randomly in order of the largest surface area. We collected image sequences and plan sequences during the task execution by operating the robot using the planner. The plan sequences include 1) the angular velocities fed into every robot joint angle at each point in time, 2) the gripper velocity, and 3) the type of actions such as gripping and placing. The image sequences were gathered by operating the planner sequence in the MuJoCo simulator [38]. When generating the images, we rendered them from a viewpoint that shows the objects from the robot’s front. In our dataset, the image resolution was set at 420×256 pixels.

Evaluation Protocols. We used our dataset to prepare 1381 scenes for training (716 successful, 665 failed), 204 scenes for validation (128 successful, 76 failed), and 816 scenes for testing (510 successful, 306 failed). About 90% of scenarios where the planner failed the task are due to mistakes in the order of stacking objects.

For evaluation metrics, we utilized the Balanced accuracy, AUC, and F1 score, which take into account class imbalances. We arranged for five different seeds and carried the evaluation using the average from the results obtained from each seed.

Implementation Details. We use Adam optimizer [19] for all experiments. When using Adam, we set the learning rate at 0.1×10^{-3} and the weight decay at 0.1×10^{-5} as in [35]. The training epoch and scheduling hyperparameter s (see Sect. 3.3) are set to 35 and 26 for the proposed framework with LSTM and 70 and 40 for the proposed framework with RSSM, respectively. The batch size is set to 8 for all experiments.

We selected a model at the highest balanced accuracy on the validation data for each seed. The input images were resized to 64×64 pixels. We used a single NVIDIA GeForce GTX 1080 Ti for training and implemented it using PyTorch [30].

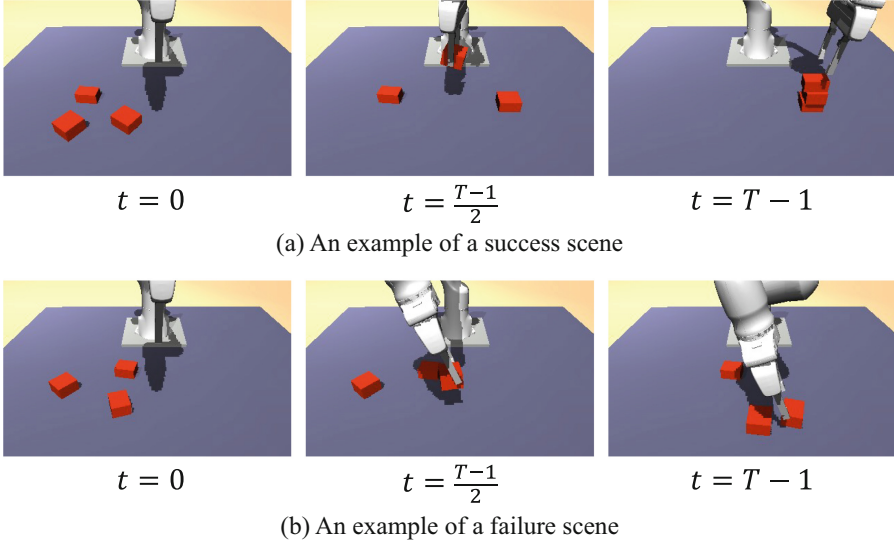


Fig. 5. Example images of the stacking task dataset. Here, (a) and (b) show an example of the success and failure scenes, respectively. In our experiments, we set up a task of stacking three cuboids placed randomly in order of the largest surface area and collected image sequences and plan sequences during the task execution, by operating the robot using the planner.

4.2 Results

We first describe the comparisons with the existing method as the main results of our experiments. Then, the ablation study and the evaluation results of the computational efficiency are described.

Main Results. We evaluated the performance of the proposed framework with RSSM and LSTM, using the image reconstruction loss described in Eq. 10, and the conventional approach, FIRP [35]. Note that in FIRP [35], the input of Classifier is not outputs of the future predictor but outputs of the image encoder. Hence, we do not apply classification weight scheduling to FIRP. The quantitative evaluation results for success-or-failure prediction on the stacking task are shown in Table 1. As shown this comparison result, applying the classification from the final state mentioned in Sect. 3.2 to the three framework, e.g. FIRP [35], RSSM, and LSTM, leads better performances than the baselines across all evaluation metrics. Furthermore, the comparisons show that the joint use of the proposed framework with the final state is superior to the existing method.

Ablation Study. We evaluate the effectiveness of each components of our framework, i.e., the new base model that solves the domain gap problem, the use

Table 1. Quantitative results for the stacking task of multiple objects. The best/second results are shown in **bold/italic**, respectively. By applying the classification from the final state to the three framework, e.g., FIRP [35], RSSM, LSTM, leads better performances than the baselines across all evaluation metrics.

	Final state	Balanced Acc.↑	AUC↑	F1↑
FIRP [35]		0.577 ± 0.047	0.594 ± 0.048	0.542 ± 0.100
	✓	0.658 ± 0.038	0.709 ± 0.053	0.631 ± 0.063
Ours w/RSSM	✓	0.918 ± 0.008	<i>0.946 ± 0.006</i>	0.914 ± 0.007
Ours w/LSTM	✓	<i>0.913 ± 0.010</i>	0.950 ± 0.004	<i>0.906 ± 0.011</i>

Table 2. Ablation study for the stacking task of multiple objects. The best/second results are shown in **bold/italic**, respectively. By applying the use of the final state to the proposed framework with RSSM or LSTM, we can see that the all evaluation measures are improved.

(a) Result for the proposed framework with RSSM [12]

	Final state	Weight scheduling	Balanced Acc.↑	AUC↑	F1↑
Ours w/ RSSM			0.897 ± 0.011	0.931 ± 0.007	0.895 ± 0.011
		✓	0.900 ± 0.012	0.940 ± 0.008	0.891 ± 0.015
	✓		<i>0.904 ± 0.014</i>	<i>0.941 ± 0.003</i>	<i>0.898 ± 0.019</i>
	✓	✓	0.918 ± 0.008	0.946 ± 0.006	0.914 ± 0.007

(b) Result for the proposed framework with LSTM [16]

	Final state	Weight scheduling	Balanced Acc.↑	AUC↑	F1↑
Ours w/ LSTM			0.856 ± 0.015	0.913 ± 0.011	0.841 ± 0.016
		✓	0.867 ± 0.008	0.922 ± 0.004	0.851 ± 0.013
	✓		<i>0.900 ± 0.011</i>	<i>0.936 ± 0.015</i>	<i>0.892 ± 0.018</i>
	✓	✓	0.913 ± 0.010	0.950 ± 0.004	0.906 ± 0.011

of the final state and the weight scheduling. The results of our ablation studies are shown in Table 2(a) and (b). Here, the evaluation results for the proposed framework with RSSM and LSTM are shown in Table 2(a) and (b), respectively. In the top row of each, our new base models show significant improvements comparing with the result of original FIRP in Table 1. By applying the use of the final state to the proposed framework with RSSM or LSTM, we can see that the all evaluation measures are improved.

Furthermore, by utilizing the classification weight scheduling mentioned in Sect. 3.3, we confirmed improvements in performance in both frameworks of classifying from the final state and classifying from all states with LSTM and RSSM.

These results clearly suggest that feeding meaningful states of future prediction into the classifier from the beginning of training with classification loss is effective for the training process of success-or-failure prediction.

Training and Inference Time and GPU Memory Usage. Finally, we evaluate the computational efficiency of the proposed method in terms of the training and inference time and GPU memory usage. The Table 3 shows the evaluation results of the training and inference time, along with the GPU memory usage for the conventional and the proposed frameworks. The proposed framework with LSTM shortened the training time by approximately 83% from FIRP [35] and lessened the GPU memory usage by around 72%. FIRP and the proposed framework with RSSM calculate image features at each step during training to predict posterior distribution. Conversely, our framework with LSTM computes image features only for the initial image during training and doesn't utilize latent overshooting [12], making it feasible to reduce training time compared to the RSSM-based methods significantly. These results show that the proposed framework is effective in terms of computational efficiency while improving the performance of classification accuracy.

Table 3. Measurement results of the total training time (hours), inference time (seconds), and maximum GPU memory usage by tensors. The best/second results are shown in **bold**/*italic*, respectively. The proposed framework with LSTM can reduce training time and inference time compared to FIRP [35].

	Training		Inference	
	Time [hours]	GPU mem. [MB]	Time [seconds]	GPU mem. [MB]
FIRP [35]	<i>2.00</i>	<i>4744</i>	<i>0.028</i>	211
Ours w/ RSSM	5.36	5386	<i>0.028</i>	<i>207</i>
Ours w/ LSTM	0.34	1318	0.027	200

5 Conclusion

We have proposed novel frameworks to predict success or failure of a task performed by a robotic arm, from the results of the future predictor utilizing a recurrent neural network. The proposed frameworks enhance the accuracy of success-or-failure prediction by deriving it from the predicted results of the final state and scheduling weight for classification loss. In our experiments on the object stacking tasks, we verified that the proposed frameworks outperform the conventional approach in all evaluation metrics regarding the accuracy of success-or-failure prediction. Compared to the conventional method, the proposed framework has shown about 83% reduction in compute time during training and approximately 72% decrease in GPU memory usage.

Acknowledgment. The authors would like to thank Hiroyuki Oyama, Kei Takaya, and Yuki Taya of NEC Corporation for their helpful comments and advices on this study.

References

- Altan, D., Sariel, S.: What went wrong? Identification of everyday object manipulation anomalies. *Intel. Serv. Robot.* **14**(2), 215–234 (2021). <https://doi.org/10.1007/s11370-021-00355-w>
- Altan, D., Sariel, S.: Clue-ai: a convolutional three-stream anomaly identification framework for robot manipulation. *IEEE Access* **11**, 48347–48357 (2023). <https://doi.org/10.1109/ACCESS.2023.3276297>
- Amos, B., et al.: Learning Awareness Models. In: *International Conference on Learning Representations* (2018)
- Chen, A.S., Nair, S., Finn, C.: Learning generalizable robotic reward functions from “in-the-Wild” human videos. In: *Robotics: Science and Systems* (2021)
- Deng, F., Jang, I., Ahn, S.: Dreamerpro: reconstruction-free model-based reinforcement learning with prototypical representations. In: *International Conference on Machine Learning*, pp. 4956–4975. PMLR (2022)
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition*. pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
- Erkent, O., Shukla, D., Piater, J.: Visual task outcome verification using deep learning. In: *International Conference on Intelligent Robots and Systems*. pp. 4821–4827 (2017). <https://doi.org/10.1109/IROS.2017.8206357>
- Furuta, D., Kutsuzawa, K., Sakaino, S., Tsuji, T.: Motion planning with success judgement model based on learning from demonstration. *IEEE Access* **8**, 73142–73150 (2020). <https://doi.org/10.1109/ACCESS.2020.2987604>
- Garrett, C.R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L.P., Lozano-Pérez, T.: Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems* **4**, 265–293 (2021)
- Gohil, P., Thoduka, S., Plöger, P.G.: Sensor fusion and multimodal learning for robotic grasp verification using neural networks. In: *International Conference on Pattern Recognition*, pp. 5111–5117 (2022). <https://doi.org/10.1109/ICPR56361.2022.9955646>
- Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: learning behaviors by latent imagination. In: *International Conference on Learning Representations* (2019)
- Hafner, D., et al.: Learning latent dynamics for planning from pixels. In: *International Conference on Machine Learning*, vol. 97, pp. 2555–2565 (2019)
- Hafner, D., Lillicrap, T.P., Norouzi, M., Ba, J.: Mastering atari with discrete world models. In: *International Conference on Learning Representations* (2020)
- Han, D., Doya, K., Tani, J.: Variational recurrent models for solving partially observable control tasks. In: *International Conference on Learning Representations* (2019)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)

17. Inceoglu, A., Aksoy, E.E., Ak, A.C., Sariel, S.: Fino-net: a deep multimodal sensor fusion framework for manipulation failure detection. In: International Conference on Intelligent Robots and Systems, pp. 6841–6847 (2021)
18. Kent, B., Łacki, M., Rossa, C.: Building a classifier model for failure modes from robot sensor readings through a modified forward stepwise algorithm. In: International Conference on Systems, Man, and Cybernetics, pp. 436–441. IEEE (2021)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. Computing Research Repository [abs/1412.6980](https://arxiv.org/abs/1412.6980) (2014)
20. Lämmle, A., Goes, M., Tenbrock, P.: Learning-based success validation for robotic assembly tasks. In: International Conference on Emerging Technologies and Factory Automation, pp. 1–4 (2022). <https://doi.org/10.1109/ETFA52439.2022.9921648>
21. Lenz, I., Lee, H., Saxena, A.: Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* **34**(4–5), 705–724 (2015)
22. Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* **37**(4–5), 421–436 (2018)
23. Moldovan, B., Moreno, P., van Otterlo, M., Santos-Victor, J., De Raedt, L.: Learning relational affordance models for robots in multi-object manipulation tasks. In: International Conference on Robotics and Automation, pp. 4373–4378 (2012). <https://doi.org/10.1109/ICRA.2012.6225042>
24. Mousavian, A., Eppner, C., Fox, D.: 6-dof graspnet: variational grasp generation for object manipulation. In: International Conference on Computer Vision, pp. 2901–2910 (2019)
25. Okada, M., Taniguchi, T.: Dreaming: model-based reinforcement learning by latent imagination without reconstruction. In: 2021 IEEE International Conference on Robotics and Automation, pp. 4209–4215. IEEE (2021)
26. Okada, M., Taniguchi, T.: Dreamingv2: reinforcement learning with discrete world models without reconstruction. In: International Conference on Intelligent Robots and System, pp. 985–991. IEEE (2022)
27. Pan, T., Wells, A.M., Shome, R., Kavraki, L.E.: A general task and motion planning framework for multiple manipulators. In: International Conference on Intelligent Robots and System, pp. 3168–3174. IEEE (2021)
28. Park, D., Hoshi, Y., Kemp, C.C.: A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robot. Autom. Lett.* **3**(3), 1544–1551 (2018). <https://doi.org/10.1109/LRA.2018.2801475>
29. Pastor, P., Kalakrishnan, M., Chitta, S., Theodorou, E., Schaal, S.: Skill learning and task outcome prediction for manipulation. In: International Conference on Robotics and Automation, pp. 3828–3834 (2011). <https://doi.org/10.1109/ICRA.2011.5980200>
30. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, pp. 8024–8035. Curran Associates, Inc. (2019)
31. Rodríguez, C., Suárez, R.: Combining motion planning and task assignment for a dual-arm system. In: International Conference on Intelligent Robots and System, pp. 4238–4243. IEEE (2016)
32. Rubinstein, R.Y.: Optimization of computer simulation models with rare events. *Eur. J. Oper. Res.* **99**(1), 89–112 (1997). [https://doi.org/10.1016/S0377-2217\(96\)00385-2](https://doi.org/10.1016/S0377-2217(96)00385-2)

33. Sak, H., Senior, A.W., Beaufays, F.: Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *Computing Research Repository abs/1402.1128* (2014). <http://arxiv.org/abs/1402.1128>
34. Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., Pathak, D.: Planning to explore via self-supervised world models. In: *International Conference on Machine Learning*, pp. 8583–8592. PMLR (2020)
35. Sogi, N., Oyama, H., Shibata, T., Terao, M.: Future predictive success-or-failure classification for long-horizon robotic tasks. *arXiv preprint arXiv:2404.03415* (2024)
36. Takano, R., Oyama, H., Yamakita, M.: Continuous optimization-based task and motion planning with signal temporal logic specifications for sequential manipulation. In: *International Conference on Robotics and Automation*, pp. 8409–8415 (2021)
37. Thoduka, S., Gall, J., Plöger, P.G.: Using visual anomaly detection for task execution monitoring. In: *International Conference on Intelligent Robots and Systems*, pp. 4604–4610 (2021). <https://doi.org/10.1109/IROS51168.2021.9636133>
38. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: *International Conference on Intelligent Robots and Systems*, pp. 5026–5033 (2012)
39. Umay, I., Fidan, B., Melek, W.: An integrated task and motion planning technique for multi-robot-systems. In: *International Symposium on Robotic and Sensors Environments*, pp. 1–7. IEEE (2019)
40. Vaswani, A., et al.: Attention is all you need. *Advances in Neural Information Processing Systems* **30** (2017)
41. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. *Neural Comput.* **31**(7), 1235–1270 (2019)
42. Zeng, A., Song, S., Yu, K.T., Donlon, E., Hogan, F.R., Bauza, M., Ma, D., Taylor, O., Liu, M., Romo, E., et al.: Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *Int. J. Robot. Res.* **41**(7), 690–705 (2022)
43. Zhang, J., Li, M., Yang, C.: Robotic grasp detection using effective graspable feature selection and precise classification. In: *International Joint Conference on Neural Networks*, pp. 1–6 (2020). <https://doi.org/10.1109/IJCNN48605.2020.9207172>
44. Zhang, X., Zhu, Y., Ding, Y., Zhu, Y., Stone, P., Zhang, S.: Visually grounded task and motion planning for mobile manipulation. In: *International Conference on Robotics and Automation*, pp. 1925–1931 (2022). <https://doi.org/10.1109/ICRA46639.2022.9812055>
45. Zhou, X., Wu, H., Rojas, J., Xu, Z., Li, S.: Nonparametric bayesian method for robot anomaly monitoring. In: *Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection*, pp. 51–93. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-6263-1_4



EGOFALLS: A Visual-Audio Dataset and Benchmark for Fall Detection Using Egocentric Cameras

Xueyi Wang^(✉) 

University of Groningen, Groningen, Netherlands
xueyi.wang@rug.nl

Abstract. Falls are significant and often fatal for vulnerable populations such as the elderly. Previous work has addressed the detection of falls by relying on data captured by single sensors, images, or accelerometers. Firstly, we collected and published a new dataset on which we assess our proposed approach. We believe this to be the first public dataset of its kind. The dataset comprises 10,948 video samples from 14 subjects. Additionally, we relied on multimodal descriptors extracted from videos captured by egocentric cameras. Our proposed method includes a late decision fusion layer that builds on top of the extracted descriptors. We conducted ablation experiments to assess the performance of individual feature extractors, the fusion of visual information, and the fusion of both visual and audio information. Moreover, we experimented with internal and external cross-validation. Our results demonstrate that the fusion of audio and visual information through late decision fusion improves detection performance, making it a promising tool for fall prevention and mitigation.

Keywords: Fall detection · Activity recognition · Multi-modality · Egocentric vision sensor fusion · Intermediate feature fusion · Late decision fusion

1 Introduction

Falls represent a major source of morbidity and mortality among the elderly population, often leading to both physical injuries and psychological consequences [1]. According to recent statistics reported by Haagsma et al. [2], approximately 14% of the adult population in Western countries have experienced fall-related injuries. Alarming, the incidence of falls within this demographic has surged by 54% since 1990 [3]. Such trends not only place a considerable burden on families caring for elderly individuals who have sustained injuries, but also have significant implications for the financial health of nations, as they strain healthcare systems and associated resources.

In the domain of fall detection, research has explored various sensing modalities, which can be categorized into four primary types: wearable sensors, fixed

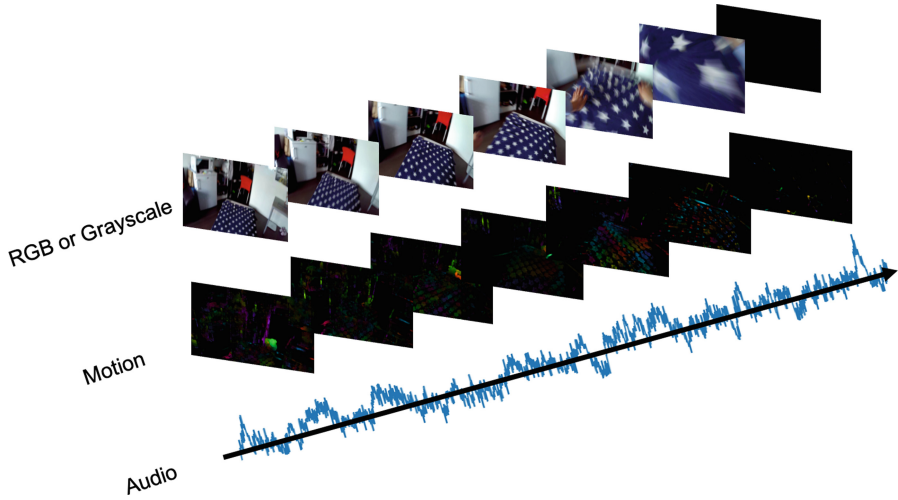


Fig. 1. Three streams of spatial, motion of vision, and audio by video clips.

visual sensors, ambient sensors, and sensor fusion [3]. Wearable sensors are a popular choice due to their portability and capability to capture data without spatial constraints. Furthermore, they offer the advantage of directly recording physiological changes associated with the human body. Fixed visual sensors, on the other hand, are valued for their simplified hardware, high-quality imaging, and reliable performance. Various forms of visual sensors have been investigated, including fixed RGB cameras and RGB-D depth cameras [4] and wearable cameras [5, 6]. The integration of wearable and visual sensors has given rise to wearable cameras as a promising technology for fall detection.

Huang et al. [7] proposed an innovative approach for egocentric visual-audio object localization, addressing the challenges posed by ego-motion and out-of-view auditory components. Their proposed method comprises a geometry-aware temporal aggregation module, a cascaded feature enhancement module, and a soft localization module. The EPIC-Fusion study by Kazakos et al. [8] introduces a novel architecture for multi-modal temporal binding, integrating RGB, Flow, and Audio modalities for the purpose of egocentric action recognition. The method achieved state-of-the-art results on the largest egocentric dataset, EPIC-Kitchens, and underscores the significance of auditory input in egocentric cameras for identifying actions and interactive objects.

In a separate study, Xiao et al. [9] presented a novel architecture termed Audiovisual SlowFast Networks (AVSlowFast) designed for integrated audiovisual perception in video recognition. The AVSlowFast architecture fuses audio and visual features at multiple levels to form a unified representation, incorporating a novel DropPathway technique to mitigate training difficulties. The study establishes the effectiveness of the AVSlowFast architecture for both supervised

and self-supervised learning of audiovisual features, offering promising prospects for advancements in video analysis applications.

Despite extensive research in fall detection, previous studies have not offered a solution tailored to low-illumination conditions. On the contrary, our study employs egocentric cameras and fuses RGB, infrared visual, and audio data to detect falls at varying illumination levels, addressing a gap that has remained unexplored in prior research. The contributions of this paper are listed as follows:

- We have assembled an extensive multimodal dataset comprising RGB and infrared videos, as well as audio recordings, for the purpose of investigating fall detection. This dataset also holds potential for application in research focused on fall-related activity recognition using egocentric cameras.
- We developed a late decision model that could detect falls successfully in both low-light and high-light environments as the benchmark for this data set.

We organize the rest of the paper as follows. We introduce existing public fall detection datasets and explain how to design data collection in Sect. 2. Then we describe the methods in Sect. 3, show the experimental results, and conduct a discussion in Sect. 4. We draw some conclusions in Sect. 5.

2 Review of Previous Dataset

2.1 Review of Previous Dataset and Research by Egocentric Cameras

Initially, we undertook a systematic literature review to identify existing datasets pertinent to fall detection research. We compiled a list of publicly used public datasets derived from various sensors in Table 1. These data sets employ a range of sensors, including wearable devices, ambient sensors, and fixed surveillance cameras, to detect falls. However, a significant shortcoming of these earlier datasets is their lack of critical elements that mimic real-world conditions, such as diverse lighting settings, subject populations, environmental conditions, and camera placements. Predominantly, these datasets were assembled in controlled laboratory environments using young, healthy individuals. This practice has resulted in pronounced homogeneity in the age, physical condition, and demographic background of the participants, which potentially limits the broader applicability of the fall detection systems developed from these data sets. However, these devices exhibit certain limitations to varying degrees and wearable cameras mitigate some of the limitations inherent in devices such as accelerometers and fixed surveillance systems to a certain extent. Wearable cameras have been the subject of extensive research across a diverse array of applications, illustrating their adaptability and profound influence. Remarkably, none of these datasets utilized egocentric cameras, which are vital for capturing a first-person perspective of falls. These devices have garnered substantial attention in multiple disciplines, reflecting their broad utility. The academic community has devoted considerable effort to exploring everyday life and social interactions through the

lens of egocentric vision. This research encompasses several key areas including activity recognition [10], object recognition [11], egocentric 3D body pose estimation [12], sentiment analysis [13], and action anticipation [14]. The significance of these studies is underscored by their high frequency of citations and their substantial contributions to various fields of research. In practical applications ranging from healthcare, sports, and security, the utility of wearable cameras in the capture of first-person visual data has been widely acknowledged. For example, in the healthcare sector, wearable cameras are employed to monitor patients with neurodegenerative diseases, evaluate rehabilitation outcomes, and enhance the efficacy of telemedicine consultations. In the realm of sports, wearable cameras facilitate detailed analyses of athletic performance and contribute to injury prevention strategies. More specifically, wearable cameras have been assessed for their utility within the elderly population across several domains, including memory assistance, support for independent living, and emergency response mechanisms. These evaluations demonstrate the potential of wearable technologies to improve the quality of life and safety of seniors. In this data collection, we take them into account. We will use cameras, the age of subjects, environments, and guidelines for data simulation to explain the specifics of data collecting. Although attempts were made by researchers such as in studies by [15, 16] to create fall detection datasets using egocentric cameras, they did not make these datasets publicly available. In contrast, our dataset, EGOFALLS, stands out as the largest collection (10948) of fall incident data recorded from egocentric cameras, featuring more instances than the 330 clips in [15] and 237 in [16]. A multimodal dataset that combines vision and audio data using egocentric vision for fall detection offers several advantages and addresses various challenges inherent in fall detection systems. To our best knowledge, EGOFALLS is the only public dataset by egocentric cameras which cover both young and elder subjects, and also the largest including 10948 clips.

2.2 EGOFALLS Dataset

In this section, we will explain the methodology employed in constructing our dataset, addressing various aspects of the data collection process and detailing the specific approaches used to ensure comprehensive coverage and robustness as follows:

Equipment: Data was collected using two types of wearable cameras: the OnReal G1 and CAMMHD Bodycams. The OnReal G1 is a compact mini action camera measuring $420 \times 420 \times 200$ mm, capable of capturing high-resolution videos up to 1080P at 30 fps. On the other hand, the CAMMHD Bodycam is a larger body camera measuring $800 \times 500 \times 300$ mm, equipped with infrared sensors for night vision. These cameras were strategically attached to various locations on the human body, such as the waist and neck, to gather comprehensive visual, motion, and audio information from different environments as shown in Fig. 1. For data collection, the chosen settings were 1080p video mode at 30 frames per second. Notably, the OnReal G1 frames include three distinct R, G, B

Table 1. Previous datasets of fall detection (A: accelerometer, G: gyroscope, M: Magnetometer, EMG: Electromyography, P: Pressure sensor, RC: RGB cameras, TC: Thermal camera)

Dataset of fall detection					
Name/Year	Egocentric	public	Sensors	Type, Duration	No. Subjects (age)
[15] (2012)	Yes	No	RC	330	3 (N/A)
[16] (2013)	Yes	No	RC	237	5 (N/A)
[17] (2014)	No	No	A, G	Sim, Short	3 (N/A)
[18] (2015)	No	No	A, G, M	Sim, Short	10 (22-29)
[19] (2015)	No	No	A, G	Sim, Short	2 (N/A)
[20] (2016)	No	No	RC, TC	Sim, Short	5 (27-81)
[21] (2016)	No	No	A, G, M	Sim, Short	5 (N/A)
[22] (2018)	No	No	A, G, M	Sim, Short	2 (N/A)
[23] (2019)	No	Yes	A, G	Sim, Short	30 (N/A)
[24] (2019)	No	Yes	A, G, M	Sim, Short	57 (20-47)
[25] (2019)	No	Yes	RC	Sim, Short	17 (18-24)
[26] (2019)	No	No	RC, TC	Sim, Short	14 (N/A)
[27] (2020)	No	No	A, G	Sim, Short	17 (N/A)
[28] (2020)	No	Yes	EMG, P	Sim, Short	12 (23-27)

channels, while CAMMHD Bodycam frames consist of three identical, repetitive grayscale channels. The resulting dataset is valuable for this work, enabling a comprehensive analysis of events and activities.

Subject: In this data collection study, 14 volunteers (19–61 years old) participated, consisting of 12 males and 2 females, with 12 young healthy individuals and two elder subjects. All participants provided their consent, acknowledging that their data would be used for research and potentially made public. The majority of subjects (11 out of 14) successfully completed data collection for four types of falls and nine types of non-falls, both indoors and outdoors. However, three subjects were unable to complete all data collection due to personal reasons. This research offers valuable insights into falls and non-falls behavior and highlights the commitment and dedication of the majority of participants to the study.

Environment: We aim to comprehensively address both indoor and outdoor environments by encompassing 14 distinct common outdoor settings and 15 diverse indoor spaces to gather data from all subjects. To enhance the variety of environmental conditions, participants are encouraged to alter their positions or directions after each activity. By incorporating this approach, we ensure a well-rounded and encompassing dataset, enabling us to draw more robust conclusions and insights for our research.

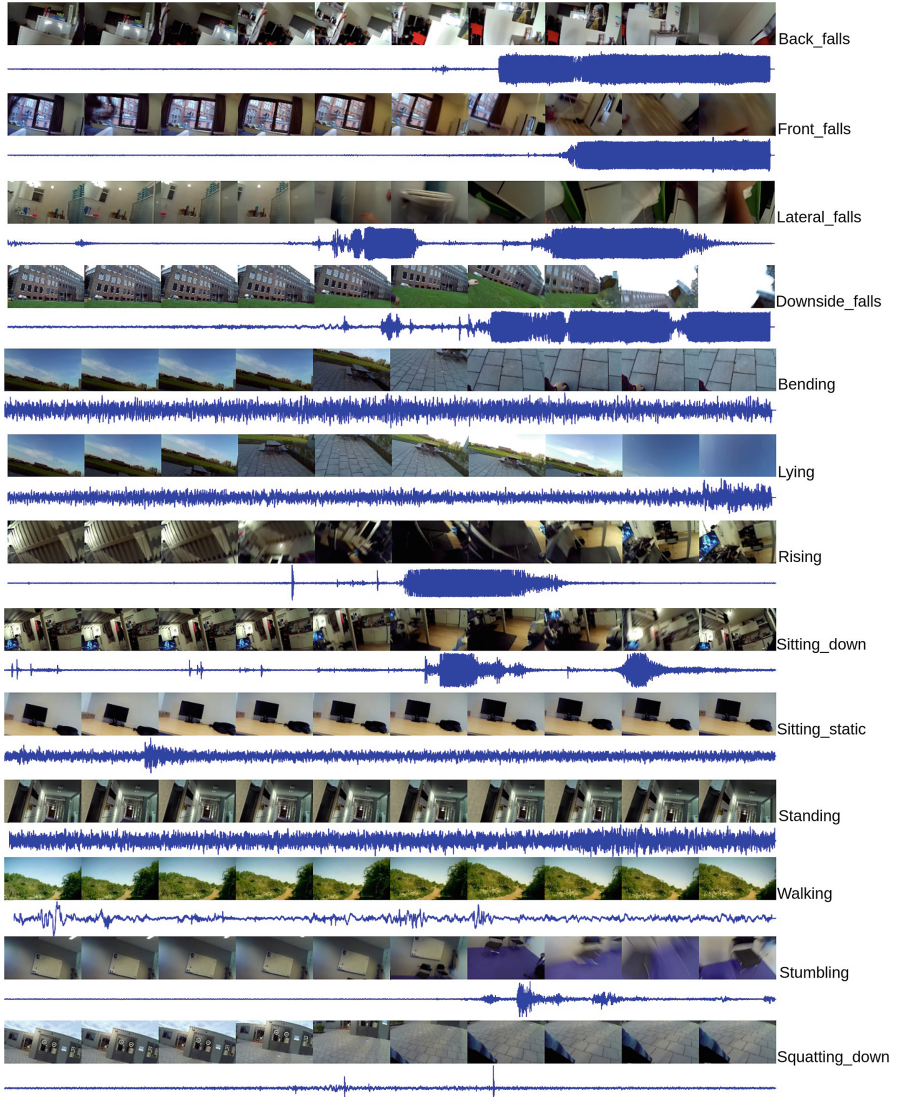


Fig. 2. Visual and audio examples of 13 activities by egocentric cameras.

Data Collection: We explore data collection from two perspectives: vision and audio. For visual information, we adhere to the guidelines provided in [29] [30]. According to [30], falls and other fall-related activities typically last 1–3 s, and they proposed a comprehensive set of trials comprising 20 types of falls with various directions and interactions with different objects. In contrast, there are no specific guidelines for audio data, as previous studies have predominantly focused on visual information. Our audio dataset comprises three categories: subject

audio, subject-object audio, and environment audio. To authentically recreate the sensory experience of falls, we expose participants to online videos depicting real-life incidents of individuals encountering genuine falls. These videos faithfully capture both the visual and auditory elements of these occurrences, allowing participants to immerse themselves in a true-to-life simulation of the event. Manual inspection of all clips helped identify common audio patterns. For falls, subject audio includes yelling, shouting, and moaning, subject-object audio captures sounds like hitting the ground or mattress, while environment audio encompasses ambient sounds such as traffic, wind, rain, animals, crowds (outdoor environment), and television, music, and talking (indoor and outdoor environments). Not every clip contains all these sounds, with some having none or a majority of them. Non-fall activities are categorized into two groups: those with strong subject-object audio (stumbling, walking, sitting-down, rising, lying), weak subject-object audio (bending, squatting down), and others without subject-object audio (sitting-static, standing). Notably, distinct sounds on the ground accompany stumbling and walking, while sounds of interacting with furniture accompany sitting down and rising. For bending and squatting, only friction sounds from cameras and clothes are noticeable. These findings provide valuable insights into audio patterns in various activities and can enhance future research in the field. The illustration of all thirteen types of activities is presented in Fig. 2. The dataset is publicly available¹ with more description in the GitHub repo². We will also publish the smaller version of the dataset either in dataverseNL or in the GitHub repo because the current dataset is too big to use and test (Table 2).

3 Methods

Visual Descriptors by Handcrafted Features. In the context of this study, we employ three types of handcrafted feature descriptors, namely Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Optical Flow. HOG, predominantly utilized in object detection tasks, quantifies the occurrences of varied gradient orientations within localized regions of a given image. The LBP descriptor [31], conversely, characterizes the neighborhood of image elements using binary codes. It captures diverse features including edges, lines, spots, and flat areas, by leveraging two complementary measures: local spatial patterns and grayscale contrast. Lastly, Optical Flow, our third handcrafted feature, measures the apparent motion between two consecutive video frames at each position, thereby providing a detailed analysis of the temporal changes in the video frames.

Truncating and Aligning Video Descriptors. We computed the similarity for three types of handcrafted feature descriptors using two vectors of size $k = 1152$, extracted from consecutive frames, resulting in a vector of $n - 1$ elements

¹ <https://dataverse.nl/dataset.xhtml?persistentId=doi:10.34894/HO5GE3>.

² <https://github.com/Xueyi-Wang/EGOFALLS>.

Table 2. Quantity and type of video clips per participant, where C1 and C2 refer to camera 1 and camera 2, and 0 means that no videos were collected for the corresponding activities.

Data	ID	Camera/Time	All	Non-Falls	Falls	Indoor	Outdoor	Waist	Neck
P1	S_H	C1-RGB/daytime	1096	328	768	554	542	548	548
P2	S_M	C1-RGB/daytime	938	426	512	562	376	469	469
P3	S_R	C1-RGB/daytime	1630	680	950	812	818	815	815
P4	S_W	C1-RGB/daytime	1298	536	762	586	712	649	649
P5	S_XL	C1-RGB/daytime	896	444	452	374	522	448	448
P6	S_Q	C1-RGB/daytime	658	498	160	346	312	329	329
P7	S_FI	C1-RGB/daytime	208	136	72	116	92	104	104
P8	S_HB	C1-RGB/daytime	490	316	174	278	212	245	245
P9	S_F	C1-RGB/daytime	142	142	0	142	0	71	71
P10	S_JF	C1-RGB/daytime	148	148	0	148	0	74	74
P11	S_L	C1-RGB/daytime	380	217	163	248	132	190	190
P12	S_D_W	C1-RGB/night	446	318	128	246	200	223	223
P13	S_D_WD	C1-RGB/night	394	264	130	186	208	197	197
P14	S_I_R	C2-Infra/night	500	500	0	196	304	250	250
P15	S_I_W	C2-Infra/night	454	336	118	230	224	227	227
P16	S_I_ZJ	C2-Infra/night	628	444	184	316	312	314	314
P17	S_I_CZ	C2-Infra/night	642	478	164	322	320	321	321
All	All	All	10948	7177	3771	5628	5320	5474	5474

for each video, each representing the cosine similarity as shown in Eq. 1. In our system, we process videos within a user-defined time window of at least 8 s, adjustable for specific needs, and from our dataset of 8 to 40-second-long video clips, we truncate each video’s descriptors to a uniform length of 238 elements, centered around the maximum value within the time window.

$$\text{cosine similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^k A_i B_i}{\sqrt{\sum_{i=1}^k A_i^2} \sqrt{\sum_{i=1}^k B_i^2}}, \quad (1)$$

Visual Descriptors By Deep Features. In this study, we harness the power of deep features extracted from video frames using the pre-trained ResNet-50 model [32], renowned for its robustness against network degradation and vanishing gradients due to its deep residual network architecture and skip connections. We resize each video frame to 224×224 pixels, extract a 2048-element vector from the network’s last fully connected layer as a global image descriptor, and, for computational efficiency, represent each clip with a 20,480-element vector by concatenating the feature descriptors from ten equally spaced frames.

Audio Features. The Mel-frequency Cepstral Coefficients (MFCC) feature extractor in the Librosa library is a widely used method for audio signal processing, transforming audio signals into compact representations that consider the characteristics of the human auditory system. This process involves converting the audio signal to the frequency domain, applying a Mel filter bank that mimics human ear perception, taking logarithms of the filter bank energies, and using discrete cosine transform to decorrelate the energies and reduce feature vector dimensionality, thus capturing perceptual content and yielding robust features against noise, pitch, and amplitude variations.

3.1 Fusion Methods

Sensor fusion is categorized into two distinct groups: fusion by homogeneous sensors and fusion by heterogeneous sensors. Homogeneous sensors are defined as those belonging to the same category or possessing similar characteristics, primarily measuring identical physical quantities. This uniformity often facilitates ease in data processing and integration, as all sensors yield data in analogous formats and scales. Conversely, heterogeneous sensors constitute a combination of various sensor types. This diversity allows for the acquisition of a wider spectrum of data, rendering these networks more adaptable for applications necessitating the monitoring of multiple parameters. The specifics and applications of each category will be elaborated upon in the subsequent subsections.

The classification of fusion strategies is primarily predicated on the condition of the input at the fusion layers, and this classification typically comprises three distinct categories: early fusion, intermediate fusion, and late fusion. In this work, we introduce a novel classification, denoted as “hybrid fusion,” which signifies a fusion strategy that exhibits characteristics of more than one of the aforementioned categories. These distinctions enable a structured analysis and evaluation of various fusion approaches employed in sensor-based systems, facilitating a comprehensive understanding of the evolving landscape of fusion techniques.

Intermediate Feature Fusion by Concatenation: A fundamental approach to fusing homogeneous data involves the technique of concatenation. Given n vectorised data input from different modalities $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ with $\mathbf{x}_i \in \mathbb{R}^{x_i}$, the concatenation of these vectors in horizontal manner

$$\mathbf{x}_{\text{h-concat}} = [\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n]$$

where $\mathbf{x}_{\text{concat}} \in \mathbb{R}^{\sum_{i=1}^n d_i}$.

Late Decision Fusion: In this study, we tested various decision fusion and applied meta-learning as follows:

$$\begin{aligned} \text{DecisionFusion}([p_1^H, \dots, p_m^H], [p_1^L, \dots, p_m^L], [p_1^O, \dots, p_m^O], \\ [p_1^D, \dots, p_m^D]) = [P_1, \dots, P_m], \end{aligned} \quad (2)$$

We employ a decision fusion technique to integrate the classification results of four independent models, each trained on distinct video descriptors encompassing handcrafted features (HOG, LBP, and optical flow), deep features (ResNet50), and audio features. The output vectors from Models 1 through 5 are amalgamated according to Eq. 2. We applied RF, SVM, and MLP for handcraft visual features, deep visual features, audio features, and late decision fusion separately in this work based on the features of each classifier and evaluation of previous work [5, 6, 33].

4 Experiments

4.1 Implementation Details

Evaluation Metrics. We evaluated the generalization ability of our machine learning models using both internal and external cross-validation techniques.

Experimental Setup. Internal cross-validation, implemented by dividing the entire dataset into k subsets or folds, was used to assess model stability and robustness within the dataset by training and validating the model k times, rotating the validation set each time, and calculating average performance metrics. In contrast, *external cross-validation* employed a leave-one-subject-out approach, simulating real-world situations where the model encounters unseen data, aiding in model selection, hyper-parameter tuning, and identifying over-fitting issues while also assessing predictive accuracy and adaptability to new instances, thus verifying the model’s overall effectiveness. Data from some participants including subjects of P9, P10, and P14 were excluded from the analysis due to incomplete data collection.

4.2 Results and Discussion

Internal Cross-validation. The results of our internal cross-validation are presented in Table 3. Our approach, fusing audio and vision, yielded the highest performance for binary classification with an accuracy of 0.978, and for 12-class classification with an accuracy of 0.850. Notably, fusing four types of individual visual feature extractors led to higher accuracy than using each visual feature descriptor independently. The performance of the audio feature extractor in binary classification was comparable to that of the fused visual feature extractors, though it slightly underperformed in the 12-class classification scenario.

External Cross-validation. In this study, the generalization of our model was evaluated using external cross-validation, where data from one subject was designated as the test set while data from all other subjects served as the training set. The late decision fusion model, which integrates vision and audio, demonstrated better performance, consistent with the findings in internal cross-validation. However, a decrease in performance relative to the internal cross-validation was observed as shown in Table 4. Despite this, the model achieved respectable accuracy scores of 0.875 for binary classification and 0.520 for 12-class classification in external cross-validation. Moreover, as some data were

Table 3. The results for six baseline models using internal cross-validation for both the 2-class (fall detection) and 12-class (daily activity recognition) problems. B1 refers to five kinds of individual features (HOG, LBP, optical flow, resnet50, audio), B2 refers to the fusion of visual features, and the proposed fusion approach “Ours” includes all features.

Fusion features	Handcrafted features	Deep model	Accuracy	
			2 classes	12 classes
B1	HOG	×	0.785 (\pm 0.01)	0.473 (\pm 0.03)
B1	LBP	×	0.855 (\pm 0.02)	0.555 (\pm 0.00)
B1	Optical flow	×	0.843 (\pm 0.01)	0.536 (\pm 0.01)
B1	Resnet50	✓	0.955 (\pm 0.01)	0.594 (\pm 0.01)
B1	Audio	×	0.952 (\pm 0.01)	0.730 (\pm 0.01)
B2	Vision	✓	0.952 (\pm 0.01)	0.742 (\pm 0.01)
Ours	All	✓	0.978 (\pm 0.01)	0.850 (\pm 0.01)

Table 4. The results for six baseline models using external cross-validation for both the 2-class (fall detection) and 12-class (daily activity recognition) problems. B1 refers to five kinds of individual features (HOG, LBP, optical flow, resnet50, audio), B2 refers to the fusion of visual features, and the proposed fusion approach “Ours” includes all features.

Fusion features	Handcrafted features	Deep model	Accuracy	
			2 classes	12 classes
B1	HOG	×	0.746 (\pm 0.05)	0.389 (\pm 0.21)
B1	LBP	×	0.804 (\pm 0.13)	0.427 (\pm 0.35)
B1	Optical flow	×	0.801 (\pm 0.07)	0.437 (\pm 0.33)
B1	Resnet50	✓	0.789 (\pm 0.14)	0.343 (\pm 0.26)
B1	Audio	×	0.886 (\pm 0.10)	0.395 (\pm 0.15)
B2	Vision	✓	0.845 (\pm 0.12)	0.496 (\pm 0.35)
Ours	All	✓	0.875 (\pm 0.15)	0.520 (\pm 0.33)

collected in a dark environment, results were reported separately for low and high illumination using RGB and infrared cameras. Specifically, for RGB in high illumination (daytime), an accuracy of 0.924 was achieved for the fusion of all features. For low illumination, the RGB cameras achieved 0.746 accuracy, while infrared cameras achieved 0.913 accuracy for fusion of vision and audio. For vision fusion, the accuracies were 0.883 for RGB in high illumination, 0.751 for RGB in low illumination, and 0.805 for infrared in low illumination. For audio, the accuracies were 0.902 for RGB in high illumination, 0.891 for RGB in low illumination, and 0.827 for infrared in low illumination. While high accuracy

was maintained across both low and high illumination for audio information, the results indicate that the vision feature extractor exhibited reduced performance in low illumination.

5 Conclusion

This work presents a new dataset and benchmark in the field of fall detection. We proposed a multimodal learning approach to detect falls in first-person view videos collected by egocentric cameras. The descriptors we explore are images, motion, and audio. The obtained results indicate that the proposed late decision fusion model, which combines visual and auditory data, has the capability for detecting falls under a wide range of lighting conditions, encompassing both high and low illumination scenarios during both daytime and nighttime. We also investigated the response times of our proposed method using NVIDIA Tesla K40 hardware. Feature extraction times per frame were recorded as follows: 0.069 s for HOG, 0.190 s for LBP, and 1.07 s for Resnet50. In our experimental setup, for a given 8-second window, feature extraction required 1.53 s for HOG, 4.56 s for LBP, 3.59 s for optical flow, and 4.90 s for Resnet50. Notably, these features can be extracted concurrently. Practically, continuous feature extraction from every frame is infeasible. A more viable approach, similar to the two-layer systems described in [16, 34], employs a low-energy sensor (such as an accelerometer or IMU) to detect initial event triggers. Subsequently, our methodology would analyze a defined time window surrounding the event to make a decision.

References

1. Yavuz, G., et al.: A smartphone based fall detector with online location support. In: International Workshop on Sensing for App Phones; Zurich, Switzerland, pp. 31–35 (2010)
2. Haagsma, J.A., et al.: Falls in older aged adults in 22 European countries: incidence, mortality and burden of disease from 1990 to 2017. *Inj. Prev.* **26**(Suppl 2), i67–i74 (2020)
3. Wang, X., Ellul, J., Azzopardi, G.: Elderly fall detection systems: a literature survey. *Front. Robot. AI* **7**, 71 (2020)
4. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE J. Biomed. Health Inf.* **18**(6), 1915–1922 (2014)
5. Wang, X., Talavera, E., Karastoyanova, D., Azzopardi, G.: Fall detection and recognition from egocentric visual data: a case study. In: International Conference on Pattern Recognition, pp. 431–443. Springer (2021)
6. Wang, X., Risi, N., Talavera Martínez, E., Chicca, E., Azzopardi, G.: Fall detection with event-based data: a case study. In: Computer Analysis of Images and Patterns: 20th International Conference, CAIP: Limassol, Cyprus, September 25–30, 2023, Proceedings, p. 2023. Springer, Part II (2023)
7. Huang, C., Tian, Y., Kumar, A., Xu, C.: Egocentric audio-visual object localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22910–22921 (2023)

8. Kazakos, E., Nagrani, A., Zisserman, A., Damen, D.: Epic-fusion: audio-visual temporal binding for egocentric action recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5492–5501 (2019)
9. Xiao, F., Lee, Y.J., Grauman, K., Malik, J., Feichtenhofer, C.: Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740* (2020)
10. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: *2011 International Conference on Computer Vision*, pp. 407–414. IEEE (2011)
11. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: *CVPR 2011*, pp. 3281–3288. IEEE (2011)
12. Tome, D., Peluse, P., Agapito, L., Badino, H.: xr-egopose: egocentric 3d human pose from an hmd camera. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7728–7738 (2019)
13. Talavera, E., Strisciuglio, N., Petkov, N., Radeva, P.: Sentiment Recognition in Egocentric Photostreams. In: Alexandre, L.A., Salvador Sánchez, J., Rodrigues, J.M.F. (eds.) *IbPRIA 2017*. LNCS, vol. 10255, pp. 471–479. Springer, Cham (2017). <https://doi.org/10.1007/978-3-319-58838-42>
14. Furnari, A., Farinella, G.M.: What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6252–6261 (2019)
15. Casares, M., Ozcan, K., Almagambetov, A., Velipasalar, S.: Automatic fall detection by a wearable embedded smart camera. In: *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, pp. 1–6. IEEE (2012)
16. Ozcan, K., Mahabalagiri, A.K., Casares, M., Velipasalar, S.: Automatic fall detection and activity classification by a wearable embedded smart camera. *IEEE J. Emerging Sel. Top. Circuits Syst.* 3(2), 125–136 (2013)
17. Hsieh, S.-L., Chen, C.-C., Wu, S.-H., Yue, T.-W.: A wrist-worn fall detection system using accelerometers and gyroscopes. In: *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, pp. 518–523. IEEE (2014)
18. Pierleoni, P., Belli, A., Palma, L., Pellegrini, M., Pernini, L., Valenti, S.: A high reliability wearable device for elderly fall detection. *IEEE Sens. J.* 15(8), 4544–4553 (2015)
19. Nukala, B.T., et al.: A real-time robust fall detection system using a wireless gait analysis sensor and an artificial neural network. In: *2014 IEEE Healthcare Innovation Conference (HIC)*, pp. 219–222. IEEE (2014)
20. Spasova, V., Iliev, I., Petrova, G.: Privacy preserving fall detection based on simple human silhouette extraction and a linear support vector machine. *Int. J. Bioautomation* 20(2) (2016)
21. Yang, K., Ahn, C.R., Vuran, M.C., Aria, S.S.: Semi-supervised near-miss fall detection for ironworkers with a wearable inertial measurement unit. *Autom. Constr.* 68, 194–202 (2016)
22. Gia, T.N., et al.: Energy efficient wearable sensor node for iot-based fall detection systems. *Microprocessors Microsyst.* 56, 34–46 (2018)
23. Chelli, A., Pätzold, M.: A machine learning approach for fall detection and daily living activity recognition. *IEEE Access* 7, 38670–38687 (2019)
24. Queralta, J.P., Gia, T.N., Tenhunen, H., Westerlund, T.: Edge-ai in lora-based health monitoring: fall detection system with fog computing and lstm recurrent neural networks. In: *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 601–604. IEEE (2019)

25. Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., Moya-Albor, E.: A vision-based approach for fall detection using multiple cameras and convolutional neural networks: a case study using the up-fall detection dataset. *Comput. Biol. Med.* **115**, 103520 (2019)
26. Ma, C., Shimada, A., Uchiyama, H., Nagahara, H., Taniguchi, R.: Fall detection using optical level anonymous image sensing system. *Optics Laser Technol.* **110**, 44–61 (2019)
27. Kerdjadj, O., Ramzan, N., Ghanem, K., Amira, A., Chouireb, F.: Fall detection and human activity classification using wearable sensors and compressed sensing. *J. Ambient. Intell. Humaniz. Comput.* **11**(1), 349–361 (2020)
28. Xi, X., Jiang, W., Lü, Z., Miran, S.M., Luo, Z.-Z.: Daily activity monitoring and fall detection based on surface electromyography and plantar pressure. *Complexity* (2020)
29. Abbate, S., Avvenuti, M., Corsini, P., Light, J., Vecchio, A.: Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: a survey. *Wireless Sensor Networks: Application-Centric Design*, pp. 147–166 (2010)
30. Yu, X.: Approaches and principles of fall detection for elderly and patient. In: *HealthCom 2008-10th International Conference on e-health Networking, Applications and Services*, pp. 42–47. IEEE (2008)
31. Ojala, T., Pietikainen, M., Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, pp. 582–585 IEEE (1994)
32. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
33. Wang, X., Talavera, E., Karastoyanova, D., Azzopardi, G.: Fall detection with a non-intrusive and first-person vision approach. *IEEE Sensors J.* (2023)
34. Ozcan, K., Mahabalagiri, A.K., Velipasalar, S.: Fall detection and activity classification using a wearable smart camera. In: *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE (2013)



Towards Unbiased Minimal Cluster Analysis of Categorical-and-Numerical Attribute Data

Yunfan Zhang¹, Xiaopeng Luo², Qingsheng Chen¹, Rong Zou³,
Yiqun Zhang^{1,3}(✉), and Yiu-ming Cheung³

¹ School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China

{3121008002,2112205080}@mail2.gdut.edu.cn

² School of Computer Engineering, Guangzhou Huali College, Guangzhou, China

³ Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China

{rongzou,ymc}@comp.hkbu.edu.hk, yqzhang@gdut.edu.cn

Abstract. Categorical and numerical attributes occur frequently in cluster analysis tasks. To bridge the information gap between the heterogeneous categorical and numerical attributes in cluster analysis, the existing approaches usually adopt prior assumptions to distance definition and cluster distribution, which unavoidably introduce bias to the clustering process. To address this issue, we propose to analyze mixed data comprising both categorical and numerical attributes by forming minimal clusters through neighborhood set theory. As the minimal clusters are the smallest cluster units that can be obtained without relying on prior assumptions, unbiased cluster analysis can be facilitated accordingly. To avoid information loss, distance and density metrics that are unified on both numerical and categorical attributes are also proposed and utilized to merge the minimal clusters hierarchically. It turns out that our proposed approach is highly interpretable, and is capable of accurately and robustly clustering data sets composed of any combination of numerical and categorical attributes. Extensive experimental evaluations demonstrate its efficacy.

Keywords: Cluster analysis · Categorical attribute · Neighborhood rough set · Mixed data · Unsupervised learning

1 Introduction

Cluster analysis is a common data analytic technique to identify cluster patterns from data sets. In real clustering tasks, numerical attributes with quantitative values and categorical attributes [1] with qualitative values are very common, where we call the data set composed of both numerical and categorical attributes mixed data. However, as shown in Fig. 1, the distance space of mixed data cannot

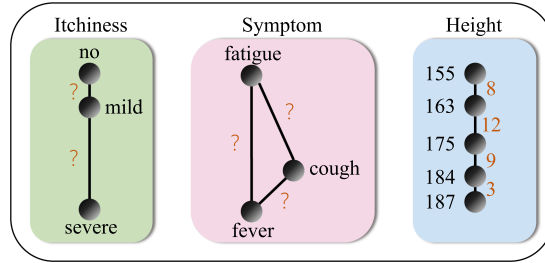


Fig. 1. Numerical attributes such as ‘Height’ can be effectively represented in Euclidean space, while quantifying dissimilarity between possible values within categorical attributes like ‘Itchiness’ and ‘Symptoms’ poses great challenges as the categorical values cannot directly participate in arithmetic operations.

be well-defined like the Euclidean distance due to the qualitative categorical data values. Additionally, the possible values of categorical attributes are usually divergent concepts in different domains with distinct implicit distance structures, which brings great challenges to the cluster analysis of mixed data.

Most existing attempts for mixed data clustering focus on the distance defining across heterogeneous attributes, and can be roughly divided into the following two streams: (i) k-ProtoType (KPT)-type methods: directly weight and combine different dissimilarity measures, e.g., Euclidean and Hamming metrics during clustering, and (ii) dedicated metric-based methods, which usually define a metric unified on the numerical and categorical attributes for distance measurement during clustering.

For KPT-type methods, the conventional KPT algorithm [11] combines Euclidean and Hamming distances [3] to cluster mixed data sets. A recent variant [17] improves the metric of categorical attributes by representing categorical values via inter-value and inter-attribute couplings, thus encoding relationships for better distance measurement. Context-based metric [13] considers attribute interdependence to form an informative categorical attribute metric. More advanced clustering methods like [16] measure the distance between possible values using Conditional Probability Distributions (CPDs) across attributes. However, these methods focus solely on proposing more advanced categorical attribute metrics and combine them with Euclidean distance for mixed data clustering, neglecting the heterogeneity of categorical and numerical attributes.

For metric-based methods, the work proposed in [7] quantifies inter-object-cluster similarity for numerical and categorical attributes within a unified probability framework, while an entropy-based approach [23] further considers the value order in categorical attributes and measures the dissimilarity degrees between different possible values from an information theory perspective. Nevertheless, these methods assume the independence of attributes, leading to information loss when applied to real-world data sets with interdependent attributes. Advanced distance definitions [14, 21] take into account attribute interdependence and preserve the corresponding information by reflecting distances based on more relevant attributes. However, they are not robust to various data sets as their effectiveness highly relies on the consistency between inherent data char-

acteristics and their assumptions, e.g., the existence of inter-value order, and inter-attribute dependence, etc.

In general, almost all the existing mixed data clustering approaches rely on certain prior knowledge or assumptions of data sets. Specifically, context-based [13] methods adopt the prior knowledge that the similarities of their possible values are reflected by the CPDs corresponding to the values obtained from other attributes, while the information theory-based methods like [23] measure dissimilarity between two possible values according to the degree of information chaos jointly demonstrated by them. Moreover, the number of true clusters is usually assumed to be known in advance. However, direct searching for oversized clusters may hinder the exploration of locally compact smaller-sized clusters. The above issues will inevitably lead to various clustering biases and thus influence clustering accuracy.

To this end, this paper proposes a universal clustering algorithm robust to various mixed data, addressing the challenges of considering heterogeneous attributes and lifting the restriction of prior knowledge. It groups data objects with distinct boundaries according to neighborhood set search, where only intra-cluster objects are expected to be included in a compact group (also called micro-cluster). Then the micro-clusters are merged to form larger “true” clusters (macro-cluster), and thus the proposed algorithm is called Mic2Mac. More specifically, a novel neighborhood relation is first proposed, forming rational and compact micro-clusters by comprehensively considering the distance and density of data objects. Subsequently, a hierarchical merging mechanism is designed to merge the current most similar micro-clusters into macro-clusters progressively. As the hierarchical merging is performed at the cluster level, the computation cost is thus not obviously increased. Extensive experiments, including comparative results, ablation studies, and visualization, affirm the superiority of Mic2Mac across various clustering methods on real benchmark data sets. The main contributions of this paper are three-fold:

- 1) A new clustering method is proposed based on neighborhood relationship to accurately form clusters of arbitrary shapes, tackling the cluster distribution bias of existing mixed data clustering methods.
- 2) An adaptive neighborhood relationship is defined based on both distance and density, leading to the generation of compact and non-overlapping micro-clusters, which has been proven to be universal and practical in the exploration of complex real-world data distributions.
- 3) Clustering process of Mic2Mac conforms to the inference process from deterministic micro-clusters to uncertain macro-clusters, providing interpretable cluster nesting relationships for multi-granular distribution analysis.

2 Related Work

As our proposed mixed data clustering approach is based on the data object partition technique, this section makes an overview of mixed data measures. It focuses on mixed data clustering methods, and data object partition techniques including k -means-type partition techniques, and neighborhood rough sets.

2.1 Mixed Data Measures

Early mixed data clustering methods like k -prototypes [11], utilized one-hot encoding to transform categorical attribute values [2] into binary vectors. However, the Hamming distance has obvious limitations in discerning differences between various value pairs. Consequently, numerous advanced techniques have emerged to efficiently address the heterogeneous attribute data, including similarity-based and representation-based approaches.

For similarity-based measures, such as context-based distance measures [13], they evaluate the distance between related attribute CPDs to highlight their dissimilarity and identify attributes with weaker dependencies. Nonetheless, these methods do not fully account for the heterogeneity of the complex categorical attributes. Subsequently, the information theory-based metric [23] measures the distances for categorical attributes by incorporating attribute weighting. Most recently, a distance learning-based approach [19] has been proposed to learn the ordinal structure of the qualitative attributes and then cluster them, while AMPHM [24] is proposed to cluster mixed data based on the rough set theory.

For representation-based measures, an interpretable representation method [16] encodes original data and further performs k -means clustering and PCA for more accurate representation. However, it is designed for categorical data only. Recently, a deep learning clustering method [5] transforms both numerical and categorical attribute values into a unified space to enable more appropriate clustering. Most recently, an approach [25] constructs minimal spanning trees for possible values to tackle qualitative-attribute clustering tasks. Moreover, the competitive theory has been utilized to handle the qualitative categorical data [4] and clustering in a federated scenario [26]. Most existing methods for clustering mixed data typically have one or both following restrictions: 1) they are tailored to data sets with one specific attribute type, and 2) they often rely on prior knowledge or assumptions.

2.2 Data Object Partition Techniques

The early k -means-type approach [12] was widely used for partitioning numerical and categorical attributes data into k clusters, while it treats all categorical variables equally during the clustering process. Recently, the representative attribute weighting partition methods w - k -means [10] was proposed for reasonably selecting variables, thereby partitioning mixed data. Nevertheless, it unreasonably assigns identical distances to different pairs of adjacent categories that may have intrinsically unequal distances, thus showing unsatisfactory partition results. Most recently, The clustering approach in [6] is proposed for incomplete data, but designed for numerical data only.

Neighborhood rough set (also called neighborhood set interchangeably for simplicity) is commonly used to partition categorical or mixed data sets. Specifically, it lets each object \mathbf{x}_i find a micro-cluster based on the neighborhood set, consisting of objects that are closer to \mathbf{x}_i . $D(\mathbf{x}_i, \mathbf{x}_j)$ represents the distance between \mathbf{x}_i and \mathbf{x}_j . The common neighborhood relations are the k -nearest

$$M^k(\mathbf{x}_i) = \{\mathbf{x}_j | D_k(\mathbf{x}_i, \mathbf{x}_j) < D(\mathbf{x}_i, \mathbf{x}_g)\}, \quad (1)$$

and the δ -radius

$$M^\delta(\mathbf{x}_i) = \{\mathbf{x}_j | D(\mathbf{x}_i, \mathbf{x}_j) \leq \delta\}, \quad (2)$$

where $j, g \in \{1, 2, \dots, n\}$, $g \neq j$, and the k in Eq. (1) represents the first k objects with the closest distance to \mathbf{x}_i . For simplification, we employ $M(\mathbf{x}_i)$ to denote the general neighborhood relation.

3 Proposed Method

In this section, we begin by formulating the problem in Sect. 3.1. Then, we present the micro partition based on the neighborhood set and the mixed data distance metric in Sect. 3.2. Finally, the hierarchical merging mechanism, and the whole clustering algorithm Mic2Mac are proposed in Sect. 3.3.

3.1 Problem Formulation

Given a mixed data set $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ comprising n data objects, each data object $\mathbf{x}_i = [x_i^1, x_i^2, \dots, x_i^d]^\top$ is a d -dimensional vector with values from the d attributes, which can also be denoted as a set $A = \{a^1, a^2, \dots, a^d\}$. The possible value set $V = \{V^1, V^2, \dots, V^d\}$ stores the value domains corresponding to each attribute. The goal of clustering is to assign the n objects to k suitable clusters $C = \{C_1, C_2, \dots, C_k\}$, where C_l is the set of data objects in the l -th cluster, with $S = \bigcup_{l=1}^k C_l$. To represent each cluster, a representative objects set $R = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k\}$ is maintained during clustering, and each representative object \mathbf{r}_l of R is a data object selected from S . A common way is to use an $n \times k$ matrix \mathbf{Q} , indicating which cluster is an object assigned to. The (i, l) -th entry $q_{i,l}$ of \mathbf{Q} is denoted as

$$q_{i,l} = \begin{cases} 1, & \text{if } l = \arg \min_g D(\mathbf{x}_i, \mathbf{r}_g), \\ 0, & \text{if } l \neq g. \end{cases} \quad (3)$$

According to Eq. (3), we have

$$\sum_{l=1}^k q_{i,l} = 1, \quad 1 \leq i \leq n, \quad (4)$$

and $q_{i,l} \in \{0, 1\}$. To appropriately cluster mixed data sets, we first need inter-object distances to be prepared where a common form can be

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{a^r \in A} D^r(x_i^r, x_j^r)^2}. \quad (5)$$

In Eq. (5), $x_i^r \in V^r$ represents the value of \mathbf{x}_i on a^r , while $D^r(x_i^r, x_j^r)$ quantifies the distance between \mathbf{x}_i and \mathbf{x}_j w.r.t. a^r . In the following subsection, we present how to define $D^r(x_i^r, x_j^r)$ to form neighborhood sets.

3.2 Micro Partition Based on Neighborhood Set

To unify the distance metric on heterogeneous attributes, we use transformation cost that quantifies the effort required to transform one Conditional Probability Distribution (CPD) into another. We begin by defining the CPD and establishing the distance between possible values of a categorical attribute to explain the principles of transformation cost quantification more clearly. Subsequently, we illustrate how this approach unifies both categorical and numerical scenarios. Finally, we derive the object-level distance and propose the micro partition based on the neighborhood set. Given a possible value v_h^r from attribute a^r , the CPD of a^t with v^t possible values $V^t = \{v_1^t, v_2^t, \dots, v_{v^t}^t\}$ is computed accordingly

$$\Psi_h^{rt} = [p(v_1^t|v_h^r), p(v_2^t|v_h^r), \dots, p(v_{v^t}^t|v_h^r)]^\top, \quad (6)$$

where $p(v_o^t|v_h^r)$ is the conditional probability of v_o^t as given v_h^r . We represent the CPD as Ψ_h^{rt} where the superscript rt signifies that this CPD characterizes the h -th possible value of a^r concerning the values of a^t . The distinction between two CPDs, such as Ψ_h^{rt} and Ψ_o^{rt} , captures the dissimilarity between v_h^r and v_o^r , according to the possible values V^t .

To quantitatively measure this dissimilarity between the CPDs describing two possible values of a categorical attribute, we employ the Earth Mover's Distance (EMD) [20], which was designed to calculate the transformation costs between two histogram descriptors. Thus, the dissimilarity between two possible values v_h^r and v_o^r , reflected by a^t can be calculated using EMD by

$$D^{rt}(v_h^r, v_o^r) = \Gamma(\Psi_h^{rt} - \Psi_o^{rt}, \mathbf{O}) \cdot \mathbf{I}, \quad (7)$$

where $\Gamma(\cdot, \cdot)$ compares each pair of corresponding bits of two vectors and retains the maximum value, while \mathbf{O} and \mathbf{I} represent a v^t -dimensional vector with all values equal to 0 and 1, respectively.

Different attributes a^t s can have varying contributions to the distance $D^{rt}(v_h^r, v_o^r)$ due to variations in inter-attribute dependence. The overall $D^{rt}(v_h^r, v_o^r)$ reflected by its respective weight w^{rt} is computed by

$$D^r(v_h^r, v_o^r) = \sum_{a^t \in A} D^{rt}(v_h^r, v_o^r) \cdot w^{rt}. \quad (8)$$

The Eq. (7) is further extended to quantify the inter-attribute dependence as the weights w^{rt} , which can be expressed as

$$w^{rt} = \frac{\sum_{h=1}^{v^r-1} \sum_{o=h+1}^{v^r} D^r(v_h^r, v_o^r)}{v^r(v^r-1)/2}, \quad (9)$$

where v^r represents the number of possible values contained within a^r . More specifically, w^{rt} measures the average transformation cost of the $v^r(v^r-1)/2$ pairs of possible values of attribute a^r reflected by a^t . According to Eqs. (7)–(9), the heterogeneous attributes are uniformly quantified as the transformation cost.

According to the work proposed in [22], the possible values of a categorical attribute are considered as concepts, so that the above process essentially quantifies the average inter-concept distances of a^r as influenced by a^t . To illustrate the principle of Eq. (8), we examine an extreme scenario. Assuming attributes a^r and a^t are identical, they will exhibit perfect interdependence, and thus their $D^{rt}(v_h^r, v_o^r)$ always reaches the maximal value, i.e., “1”, for any combinations of h and o with $h \neq o$, according to Eq. (7). Consequently, w^{rt} also reaches the maximal value of “1”, representing 100% dependence of two attributes. By applying Eqs. (7)–(9), we can obtain the distance between data objects \mathbf{x}_i and \mathbf{x}_j .

The defined dissimilarity measure applies to both categorical attributes and numerical attributes, as Eq. (8) provides a uniform treatment of heterogeneous attributes. Then we prove that our measure is a distance metric.

Theorem 1. $D(\mathbf{x}_i, \mathbf{x}_j)$ is a distance metric.

Proof. As Eq. (7) satisfies the properties of a metric, it follows naturally Eq. (8), which is derived from Eq. (7), is also a metric. Moreover, the calculation of Eq. (5) involves finite arithmetic processes according to Eq. (8), guaranteeing that $D(\mathbf{x}_i, \mathbf{x}_j)$ adheres to all essential metric properties for any $i, j, h \in \{1, 2, \dots, n\}$, which are listed as follows:

- (1) $D(\mathbf{x}_i, \mathbf{x}_j) \geq 0$; $D(\mathbf{x}_i, \mathbf{x}_j) = 0$ iff $\mathbf{x}_i = \mathbf{x}_j$;
- (2) $D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_j, \mathbf{x}_i)$;
- (3) $D(\mathbf{x}_i, \mathbf{x}_j) \leq D(\mathbf{x}_i, \mathbf{x}_h) + D(\mathbf{x}_h, \mathbf{x}_j)$. □

The conventional neighborhood sets $M^k(\mathbf{x}_i)$ and $M^\delta(\mathbf{x}_i)$ (i.e., Eqs. (1) and (2)) generate n neighborhood sets, which may partially overlap with surrounded ones, causing laborious computation with a large n . Additionally, these neighborhood relations may group dissimilar objects in the uneven distribution of data objects. To better partition objects and reduce computational costs, we have developed a new approach called micro partition based on the neighborhood set, which considers both distance and density. This approach creates non-overlapping neighborhood sets by selecting representative objects and grouping their corresponding neighbors based on *merging interval*.

Definition 1. *Merging interval:* Given an object \mathbf{x}_i with a density ρ_i , the merging interval ϕ_i signifies the minimum distance between \mathbf{x}_i and \mathbf{x}_j with a higher corresponding density ρ_j , which can be expressed as:

$$\phi_i = \min D(\mathbf{x}_i, \mathbf{x}_j), \text{ s.t. } \rho_i < \rho_j \text{ and } \mathbf{x}_j \in S \setminus \mathbf{x}_i, \quad (10)$$

where $S \setminus \mathbf{x}_i$ is the data set that excludes \mathbf{x}_i , while ρ_i and ρ_j denote the densities of \mathbf{x}_i and \mathbf{x}_j , respectively. Furthermore, for the object with the maximum density, its merging interval is defined as $\max D(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{x}_j \in S$.

In Definition 1, the density ρ_i can be computed as

$$\rho_i = \frac{D(\mathbf{x}_i, \mathbf{x}_{\langle i, q_i \rangle})}{q_i}. \quad (11)$$

Equation (11) computes the distance corresponding to the ranking q_i of the adjacent object $\mathbf{x}_{\langle i, q_i \rangle}$, which can be seen as the density of \mathbf{x}_i . Assuming the \mathbf{x}_i is a center point, $\mathbf{x}_{\langle i, q_i \rangle}$ is the q_i -th closest object to \mathbf{x}_i in n objects. Specifically, we initially create the neighbor set $AM_i = \{\mathbf{x}_{\langle i, 0 \rangle}, \mathbf{x}_{\langle i, 1 \rangle}, \mathbf{x}_{\langle i, 2 \rangle}, \dots, \mathbf{x}_{\langle i, n-1 \rangle}\}$ in *ascending* sequence relative to \mathbf{x}_i , where $\mathbf{x}_{\langle i, 0 \rangle} \equiv \mathbf{x}_i$ and $AM_i(y) = \mathbf{x}_{\langle i, y \rangle}$. Afterwards, when we iterate through AM_i from small to large, we choose the object $\mathbf{x}_{\langle i, g \rangle}$ that first satisfies the condition $D(\mathbf{x}_i, \mathbf{x}_{\langle i, g \rangle})/g < D(\mathbf{x}_{\langle i, g-1 \rangle})/(g-1)$, which confirms the value of q_i as $q_i = g - 2$. The density calculation effectively selects neighboring objects, ensuring that objects beyond a noticeable interval boundary are not included in the neighborhood set corresponding to \mathbf{x}_i . Hence, it will partition objects into compact clusters, which contain the most similar objects.

To select the most suitable representative object for a micro-cluster, we prioritize objects with higher density than their neighbors and positioning far from other representative objects. According to Definition 1, objects with greater merging intervals are considered more suitable to be the representative objects. Thus, we rank data objects based on their merging intervals in *descending* sequence and form micro-clusters based on neighborhood set by

$$M^\phi(\mathbf{x}_i) = \left\{ \bigcup_{j=1}^{q_i} AM_i(j) \right\} \setminus \left\{ \bigcup_{\phi_p > \phi_i} M^\phi(\mathbf{x}_p) \right\}, \quad (12)$$

where q_i is the q_i -th closeness to \mathbf{x}_i among all the n objects, as mentioned in Eq. (11), while the objects \mathbf{x}_p with larger merging intervals than the \mathbf{x}_i will be excluded from $M^\phi(\mathbf{x}_i)$ corresponding to \mathbf{x}_i . The process of forming micro-clusters will continue until all objects are contained by these micro-clusters. All the representative objects in each micro-cluster are stored in the micro representative objects set $MR = \{\mathbf{b}_1, \mathbf{b}_1, \dots, \mathbf{b}_m\}$, where m is the number of representative objects.

To illustrate our calculation and merging processes more clearly, Fig. 2 provides a toy example shown in processes 1–4. The proposed micro partition based on the neighborhood set is outlined in Algorithm 1. The mechanism for merging $M^\phi(\mathbf{x}_i)$ is crucial and will be discussed in the next subsection.

3.3 Merge Micro-Clusters Into Macro-Clusters

Based on our proposed micro partition, a hierarchical merging mechanism is presented to merge micro-clusters.

Given the data set S and the number of clusters k , we iteratively compute micro-clusters $M^\phi(\mathbf{x}_i)$ and update data set S at each layer in the following two steps: 1) fix S , compute $M^\phi(\mathbf{x}_i)$ and micro representative object set MR by Algorithm 1 according to dissimilarity matrices D , and 2) fix MR , update S based on MR . Specifically, the hierarchical merging mechanism utilizes MR from the previous layer as the new local data set in the next layer. This process enables multiple partitioning and merging of objects while preserving the local micro-clusters. These two steps iterate until $m = k$, where m is the number of

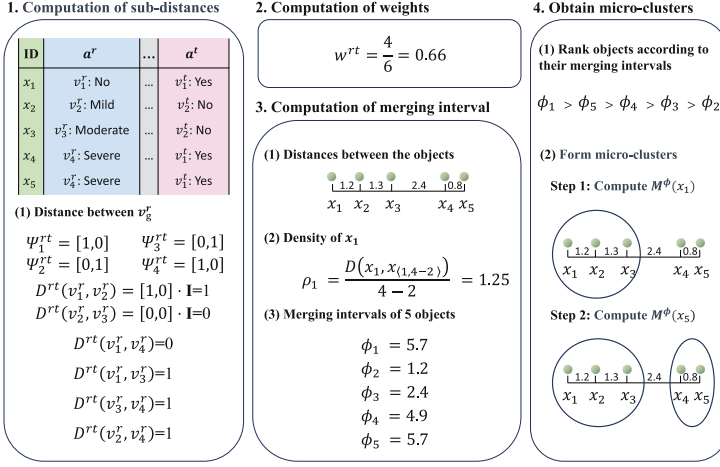


Fig. 2. A toy example illustrates the calculation processes. In processes 1 and 2, we compute the sub-distances in attribute a^r . Then, we compute the contribution of a^t to a^r . In processes 3 and 4, we confirm q_i , where $AM_1 = \{\mathbf{x}_{(1,0)}, \mathbf{x}_{(1,1)}, \mathbf{x}_{(1,2)}, \mathbf{x}_{(1,3)}, \mathbf{x}_{(1,4)}\}$. After obtaining the merging intervals corresponding to each object, we then merge data objects into micro-clusters according to the descending order of the merging intervals.

Algorithm 1. MPNS: Micro Partition based on Neighborhood Set

Input: S, D .

Output: $M^\phi(\mathbf{x}_i), MR$.

```

1: for  $i = 1$  to  $n$  do
2:   Update the density  $\rho_i$  of  $\mathbf{x}_i$  based on Eq. (11);
3: end for
4: for  $i = 1$  to  $n$  do
5:   Update the merging interval  $\phi_i$  of  $\mathbf{x}_i$  based on Eq. (10);
6: end for
7: for  $i = 1$  to  $n$  do
8:   if  $\rho_i > 0$  then
9:     Select  $\mathbf{x}_i$  as representative object  $\mathbf{b}_i$  to  $MR$ ;
10:  end if
11:  Update  $M^\phi(\mathbf{x}_i)$  based on Eq. (12);
12: end for

```

representative objects. The overall Mic2Mac clustering algorithm is outlined in Algorithm 2.

Theorem 2. *The time complexity of Mic2Mac is $O(d^2n + n \log n)$ for each iteration.*

Proof. In the worst-case scenario, all attributes are categorical, and V is equal to the maximum number of possible values across all the categorical attributes.

Algorithm 2. Mic2Mac: Merge Micro-Clusters into Macro-Clusters**Input:** S, k .**Output:** \mathbf{Q} .

```

1: Initialize the iteration counter by  $\tau = 0$ ; Set each object as a micro-cluster;
2: while  $|MR^{(\tau)}| > k$  do
3:   Update  $D$  based on Eq. (5);
4:   Update  $M^{\phi,(\tau)}(\mathbf{x}_i)$  and  $MR^{(\tau)}$  by Algorithm 1;
5:   Update  $S^{(\tau+1)}$  by  $S^{(\tau+1)} = MR^{(\tau)}$ ;
6:   Update the iteration counter by  $\tau = \tau + 1$ ;
7: end while
8: Update  $R = MR^{(\tau)}$ ;
9: Compute  $\mathbf{Q}^{(\tau)}$  according to Eq. (3).

```

To analyze the overall complexity, we compute the complexity of $D^{(\tau)}$, $M^{\phi,(\tau)}$, and hierarchical merging once, respectively.

To compute the dissimilarity matrices $D^{(\tau)}$, we need to derive $d \times d$ pairs of CPDs by scanning n data objects in data set S . This results in a $O(d^2n)$ complexity. For computing the distances between a pair of intra-attribute possible values, it takes $O(V)$ complexity for every attribute. Thus, obtaining $D^{(\tau)}$ incurs a complexity of $O(nd^2 + V)$.

Given $D^{(\tau)}$ obtained from Algorithm 2, to compute $M^{\phi,(\tau)}$, we need to sort an $n \times n$ matrix, taking $O(n + n \log n)$ complexity. Subsequently, we sort n merging intervals in $O(n \log n)$ complexity. Therefore, computing $M^{\phi,(\tau)}$ takes $O(n + 2n \log n)$.

To implement hierarchical merging, we need to update S in each iteration according to the micro representative objects set MR , which takes $O(n)$.

Therefore, the overall complexity of Mic2Mac at a given iteration τ can be simplified to $O(d^2n + n \log n)$. \square

4 Experiments

4.1 Experimental Settings

This section presents three types of experiments to comprehensively evaluate the clustering performance of our proposed Mic2Mac: (1) Clustering performance evaluation, (2) Ablation study, and (3) Visualization of cluster discrimination ability. Counterparts, validity indices, and data sets are introduced below.

Ten counterparts are compared, including Jia's Distance Metric (JDM) [14], Coupled Similarity Metric (CSM) [15], Entropy-based Distance Metric (EDM) [23], and Zhang's Distance Metric (ZDM) [21] incorporated with the conventional k -modes (KMD) [12] and k -prototypes (KPT) [11] approaches based on the attribute composition of data sets. Cheung's Iterative Learning (CIL) [7], designed for data sets with numerical and categorical attributes, is also selected. JDM, CSM, EDM, and ZDM represent state-of-the-art methods. Additionally, three conventional clustering algorithms, namely Attribute Weighting k -means

Table 1. Summary of nine utilized data sets. The columns “Categorical”, “Numerical”, “Objects”, and “Clusters” are the numbers of categorical attributes, numerical attributes, data objects, and clusters, respectively.

No.	Data Set	Abbrev.	Categorical	Numerical	Objects	Clusters
1	Dermatology	Derm	33	1	366	6
2	Autism-Adolescent	Autism	2	7	104	2
3	Common Toad	Toad	12	2	189	2
4	Hayes-Roth	Hayes	4	0	132	3
5	Breast Cancer	Cancer	9	0	286	2
6	Lymphography	Lym	18	0	148	4
7	Congressional Voting	Vote	16	0	435	2
8	Employee Selection	Employee	4	0	488	9
9	Social Workers	Workers	10	0	1000	4

Table 2. Clustering performance evaluated by CA, where the best results are highlighted in **bold** and the second-best results are underlined.

Methods	Derm	Autism	Toad	Hayes	Cancer	Lym	Vote	Employee	Workers
KMD	0.554±0.10	0.545±0.11	0.548±0.03	0.364±0.01	0.519±0.02	0.453±0.04	0.864±0.00	0.367±0.03	0.392±0.03
KPT	0.554±0.10	0.530±0.03	0.530±0.02	0.364±0.01	0.519±0.02	0.453±0.04	0.864±0.00	0.367±0.03	0.392±0.03
WKM	0.623±0.09	0.525±0.02	0.523±0.03	0.408±0.05	0.584±0.09	0.439±0.05	0.857±0.07	0.368±0.03	0.375±0.03
CIL	0.675±0.10	0.519±0.03	0.506±0.00	0.376±0.04	0.541±0.06	0.500±0.04	0.881±0.00	0.384±0.04	0.373±0.03
JDM	0.665±0.10	0.579±0.05	0.522±0.02	0.375±0.02	0.582±0.10	0.473±0.04	0.868±0.00	0.351±0.03	0.334±0.03
CSM	0.602±0.14	0.524±0.03	0.526±0.02	0.405±0.04	0.528±0.04	0.419±0.05	0.865±0.01	0.402±0.04	0.331±0.03
EDM	0.587±0.10	0.558±0.03	0.537±0.03	0.407±0.03	0.530±0.02	0.452±0.04	0.832±0.10	0.366±0.02	0.332±0.01
ZDM	0.685±0.11	0.558±0.02	0.578±0.02	0.404±0.03	0.569±0.19	0.470±0.04	0.872±0.00	0.368±0.03	0.374±0.03
Mic2Mac	0.768±0.00	0.596±0.00	0.545±0.00	0.417±0.00	0.766±0.00	0.561±0.00	0.874±0.00	0.393±0.00	0.435±0.00

(WKM) clustering algorithm [10], the original KMD, and KPT adopting Hamming and Euclidean distance metrics, are also included in the comparison. Furthermore, two variations of Mic2Mac, named Mic2-Mac^I and Mic2Mac^{II}, are introduced for ablation studies, and additional details about these two Mic2Mac variants are provided in Sect. 4.3.

Two validity indices have been chosen for comprehensively verifying the clustering performance, including CA [9] with a value range of [0, 1], and ARI [8] with a value range of [-1, 1]. A higher value for both these indices indicates better clustering performance.

Nine real-world data sets from various domains, including medicine, biology, sociology, etc., have been selected, which are shown in Table 1. Data sets 1–7 are public data sets collected from the UCI machine learning library¹. Data sets 8 and 9 are obtained from the Weka website². All data sets are pre-processed by removing objects with missing values.

¹ <https://archive.ics.uci.edu/>

² <https://waikato.github.io/weka-wiki/datasets/>

Table 3. Clustering performance evaluated by ARI, where the best results are highlighted in **bold** and the second-best results are underlined.

Methods	Derm	Autism	Toad	Hayes	Cancer	Lym	Vote	Employee	Workers
KMD	0.396±0.15	−0.003±0.01	−0.002±0.02	−0.012±0.00	−0.004±0.00	0.113±0.04	0.530±0.00	0.162±0.02	0.057±0.02
KPT	0.422±0.12	−0.003±0.01	−0.008±0.01	−0.012±0.00	−0.004±0.00	0.113±0.04	0.530±0.00	0.162±0.02	0.057±0.02
WKM	0.509±0.09	−0.006±0.01	−0.008±0.01	0.007±0.02	0.040±0.07	0.085±0.04	0.527±0.11	0.172±0.03	0.046±0.02
CIL	0.606±0.10	−0.007±0.01	−0.021±0.00	−0.004±0.02	0.011±0.04	0.182±0.05	0.579±0.00	0.193±0.02	0.052±0.02
JDM	0.614±0.13	<u>0.018±0.03</u>	−0.014±0.01	−0.006±0.01	0.041±0.07	0.123±0.04	0.541±0.01	0.167±0.02	0.052±0.01
CSM	0.518±0.17	−0.009±0.01	−0.008±0.01	<u>0.008±0.02</u>	0.003±0.02	0.089±0.04	0.532±0.03	0.212±0.03	0.051±0.02
EDM	0.439±0.12	0.006±0.01	<u>0.002±0.01</u>	<u>0.008±0.02</u>	0.007±0.01	0.089±0.03	0.478±0.17	0.163±0.04	0.059±0.01
ZDM	<u>0.627±0.15</u>	−0.015±0.01	0.013±0.02	0.007±0.02	<u>0.062±0.02</u>	<u>0.132±0.05</u>	0.553±0.01	<u>0.211±0.02</u>	<u>0.076±0.01</u>
Mic2Mac	0.678±0.00	0.019±0.00	−0.001±0.00	0.009±0.00	0.109±0.00	0.129±0.00	<u>0.557±0.00</u>	0.173±0.00	0.085±0.00

Table 4. Ave. Rank of CA and ARI rows report the average performance ranks, where the best results are highlighted in **bold**, while the second-best results are underline.

Ave. Rank	KMD	KPT	WKM	CIL	JDM	CSM	EDM	ZDM	Mic2Mac
Ave. Rank @ CA	5.944	6.389	5.278	4.889	5.222	6.111	6.167	<u>3.556</u>	1.444
Ave. Rank @ ARI	6.611	6.722	6.389	4.722	4.833	5.556	5.222	<u>3.056</u>	1.889

4.2 Clustering Performance Evaluation

The clustering performance is reported in Tables 2 and 3, which are accessed by CA and ARI, respectively. The average ranks of the CA and ARI performances across all data sets for the compared methods are presented in Table 4, based on the results in Tables 2 and 3.

The key observations are as follows: (1) Mic2Mac consistently performs the best on most data sets in terms of CA index. (2) On certain data sets, such as Toad, Vote, and Employee, Mic2Mac does not achieve the best result, but the performance gaps between Mic2Mac and the best-performing counterparts are tiny, also highlighting the superiority of Mic2Mac. (3) While Mic2Mac does not yield the best results in terms of the ARI on some data sets, e.g., Lym and Employee, it consistently performs the best and the second-best on most data sets, which still verifies its effectiveness. Intuitively, if a data set contains only numerical attributes, the performance of Mic2Mac downgrades to traditional k -means. The more categorical attributes a data set contains, the better the Mic2Mac can perform. Meanwhile, Mic2Mac also performs well on mixed data.

4.3 Ablation Study

In ablation studies, we focus on the clustering performance assessed by the ARI. Firstly, to assess the effectiveness of the dissimilarity metric proposed for heterogeneous attributes, we restrict Mic2Mac to utilize the combination of Hamming distance and Euclidean distance to tackle mixed data, forming Mic2Mac^I. Secondly, to evaluate the effectiveness of our proposed hierarchical merging mechanism, we compare Mic2Mac and Mic2Mac^I with their variation Mic2Mac^{II}, which incorporates the partitioning strategy of KPT by partitioning the representative

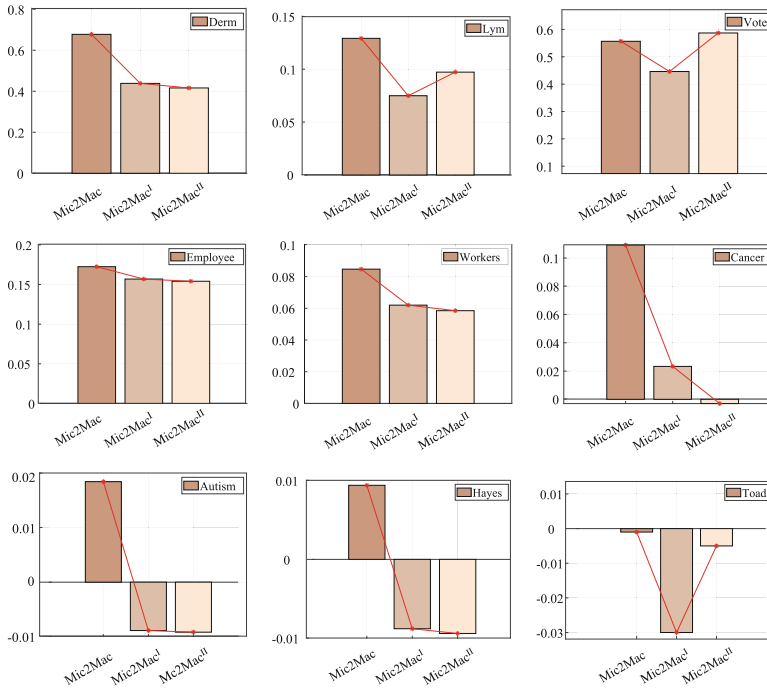


Fig. 3. Comparison of clustering performance among Mic2Mac, Mic2Mac^I, and Mic2Mac^{II} on all the 9 data sets. A better measure has a higher value. The Ave. Rank @ ARI of Mic2Mac, Mic2Mac^I, and Mic2Mac^{II} are 1.111, 2.333, and 2.556, respectively.

objects after the first formation of the micro-clusters. The clustering performance and the average rank of Mic2Mac with its two variations are illustrated in Fig. 3.

The overall result reveals that Mic2Mac consistently outperforms its two variations, demonstrating the effectiveness of Mic2Mac. Specifically, Mic2Mac surpasses Mic2Mac^I on nine data sets, indicating that Mic2Mac can effectively measure the original heterogeneous attribute data information. Furthermore, Mic2Mac outperforms Mic2Mac^{II} on eight data sets, and Mic2Mac^I performs better than Mic2Mac^{II} on six data sets. This emphasizes the effectiveness of the proposed merging mechanism. The reason why Mic2Mac^I perform worse than Mic2Mac^{II} on certain data sets (i.e. Toad, Lym, and Vote) would be that Mic2Mac^I employs the simplest Euclidean and Hamming distance measures, which makes it hard to handle the complex issues in real-world data distribution, e.g., overlapping, and coupling categorical attributes.

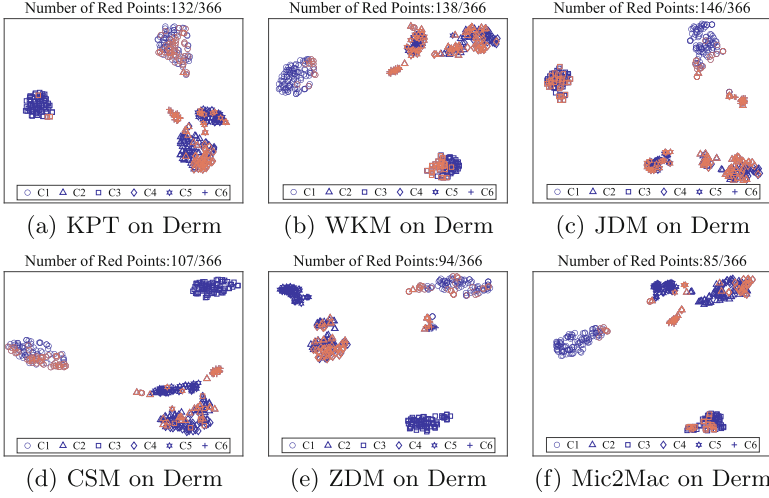


Fig. 4. t-SNE visualization of the Derm data sets, where data points marked in “C1” to “C6” indicate their “true” cluster labels, while objects marked in red indicate they were incorrectly clustered.

4.4 Visualization

In Fig. 4, t-SNE [18] is employed to showcase the cluster discrimination ability of Mic2Mac. The Derm data set is first clustered using KPT, WKM, JDM, CSM, ZDM, and Mic2Mac. Subsequently, the data is encoded according to the distance matrix of objects created by the distance metrics of the corresponding approaches, respectively. These distance matrices are treated as the representations of the data and are then projected into two-dimensional space using t-SNE for visualization. Data points are marked with different markers to indicate their “true” cluster labels. The red markers are utilized to indicate the objects that are incorrectly clustered. Intuitively, fewer red markers indicate a more accurate clustering performance and a more separable distribution of different markers indicates a more powerful cluster discrimination ability.

The visualization in Fig. 4 clearly shows that Mic2Mac exhibits fewer red markers and a more separable distribution of different markers, signifying its stronger cluster discrimination ability than the conventional and state-of-the-art methods.

5 Concluding Remarks

In this paper, a novel approach called Mic2Mac has been proposed for mixed data clustering, which simultaneously tackles two challenges inherent in clustering real-world mixed data sets, i.e., the information gap of heterogeneous attributes

and the bias brought by prior knowledge. To address these challenges, we have proposed: (1) A heterogeneous attribute metric for preserving and leveraging original data information; (2) A micro partition approach based on neighborhood set theory for forming unbiased micro-clusters; and (3) A merging mechanism for hierarchically merging micro-clusters into sought number of clusters. The superiority of Mic2Mac is evidenced through extensive experiments. Moreover, the clustering process of Mic2Mac is highly interpretable due to the nested relationship among multi-granular clusters extracted during the merging phase.

In the future, this research will be extended to address more challenging clustering analysis tasks, e.g., federated mixed data clustering, and exploring cluster patterns for unstructured multi-modal data. Moreover, the potential of the dendrogram formed by merging the micro-clusters will also be explored for understanding complex data sets.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants: 62476063 and 62102097, the NSFC/Research Grants Council (RGC) Joint Research Scheme under the grant N_HKBU214/21, the Natural Science Foundation of Guangdong Province under grant 2023A15150-12855, the General Research Fund of RGC under grants: 12201321, 12202622, and 12201323, and the RGC Senior Research Fellow Scheme under grant SRFS2324-2S02.

References

1. Agresti, A.: *Categorical Data Analysis*. Wiley Series in Probability and Statistics. Wiley (2002)
2. Alamuri, M., Surampudi, B.R., Negi, A.: A survey of distance/similarity measures for categorical data. In: *The 24th International Joint Conference on Neural Networks*, pp. 1907–1914. IEEE (2014)
3. Arabie, P., Baier, N.D., Critchley, C.F., Keynes, M.: *Studies in classification, data analysis, and knowledge organization*. Springer (2006)
4. Cai, S., Zhang, Y., Luo, X., Cheung, Y.m., Jia, H., Liu, P.: Robust categorical data clustering guided by multi-granular competitive learning. In: *The IEEE 44th International Conference on Distributed Computing Systems*, pp. 288–299 (2024)
5. Chen, J., Ji, Y., Zou, R., Zhang, Y., Cheung, Y.m.: QGRL: Quaternion graph representation learning for heterogeneous feature data clustering. In: *The 30th SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1–10 (2024)
6. Cheng, M., You, X.: Leachable component clustering. In: *The 26th International Conference on Pattern Recognition*, pp. 1858–1864 (2022)
7. Cheung, Y.m., Jia, H.: Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recogn.* **46**(8), 2228–2238 (2013)
8. Gates, A.J., Ahn, Y.Y.: The impact of random models on clustering similarity. *J. Mach. Learn. Res.* **18**, 3049–3076 (2017)
9. He, X., Cai, D., Niyogi, P.: Laplacian Score for Feature Selection. In: *The 17th Advances in Neural Information Processing Systems*, pp. 507–514 (2005)
10. Huang, J., Ng, M., Rong, H., Li, Z.: Automated variable weighting in k-means type clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 657–668 (2005)

11. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: The 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 21–34 (1997)
12. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Disc.* **2**(3), 283–304 (1998)
13. Ienco, D., Pensa, R.G., Meo, R.: From context to distance. *ACM Trans. Knowl. Discov. Data* **6**(1), 1–25 (2012)
14. Jia, H., Cheung, Y.m., Liu, J.: A new distance metric for unsupervised learning of categorical data. *IEEE Trans. Neural Networks Learn. Syst.* **27**(5), 1065–1079 (2016)
15. Jian, S., Cao, L., Lu, K., Gao, H.: Unsupervised coupled metric similarity for Non-IID categorical data. *IEEE Trans. Knowl. Data Eng.* **30**(9), 1810–1823 (2018)
16. Jian, S., Pang, G., Cao, L., Lu, K., Gao, H.: CURE: flexible categorical data representation by hierarchical coupling learning. *IEEE Trans. Knowl. Data Eng.* **31**(5), 853–866 (2019)
17. Qian, Y., Li, F., Liang, J., Liu, B., Dang, C.: Space structure and clustering of categorical data. *IEEE Trans. Neural Networks Learn. Syst.* **27**(10), 2047–2059 (2016)
18. Van Der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(86), 2579–2605 (2008)
19. Wang, P., Zhang, Y., Zhang, Y., Lu, Y., Li, M., Cheung, Y.m.: Clustering by learning the ordinal relationships of qualitative attribute values. In: The 34th International Joint Conference on Neural Networks, pp. 1–8 (2024)
20. Xu, J., Lei, B., Gu, Y., Winslett, M., Yu, G., Zhang, Z.: Efficient similarity join based on earth mover’s distance using MapReduce. *IEEE Trans. Knowl. Data Eng.* **27**(8), 2148–2162 (2015)
21. Zhang, Y., Cheung, Y.M.: A new distance metric exploiting heterogeneous interattribute relationship for ordinal-and-nominal-attribute data clustering. *IEEE Trans. Cybern.* **52**(2), 758–771 (2022)
22. Zhang, Y., Cheung, Y.M.: Graph-based dissimilarity measurement for cluster analysis of any-type-attributed data. *IEEE Trans. Neural Networks Learn. Syst.* **34**(9), 6530–6544 (2023)
23. Zhang, Y., Cheung, Y.M., Tan, K.C.: A unified entropy-based distance metric for ordinal-and-nominal-attribute data clustering. *IEEE Trans. Neural Networks Learn. Syst.* **31**(1), 39–52 (2020)
24. Zhang, Y., Zou, R., Zhang, Y., Zhang, Y., Cheung, Y.M., Li, K.: Adaptive micro partition and hierarchical merging for accurate mixed data clustering. *Complex Intell. Syst.*, 1–13 (2024)
25. Zhao, M., Feng, S., Zhang, Y., Li, M., Lu, Y., Cheung, Y.M.: Learning order forest for qualitative-attribute data clustering. In: The 27th European Conference on Artificial Intelligence, pp. 1–8 (2024)
26. Zou, R., Zhang, Y., Zhang, Y., Lu, Y., Li, M., Cheung, Y.M.: Federated clustering with unknown number of clusters. In: The 6th International Conference on Data-driven Optimization of Complex Systems. pp. 1–6 (2024)



PolSAR Image Classification Using Complex-Valued Squeeze and Excitation Network

Shradha Makhija¹, Srimanta Mandal^{1(✉)}, Utkarsh Pandya¹, Sanid Chirakkal²,
and Deepak Putrevu²

¹ DA-IICT, Gandhinagar, India

{shradha_makhija, srimanta_mandal, utkarsh_pandya}@daiict.ac.in

² SAC, ISRO, India

{sanid, dputrevu}@sac.isro.gov.in

Abstract. We aim to classify terrain into different ground covers, such as urban, crops, forests, water, etc., from polarimetric SAR (PolSAR) images. State-of-the-art classification approaches relish the advantage of deep learning techniques. However, conventional techniques, such as convolutional neural networks (CNN), developed for optical images are not quite suitable for complex-valued PolSAR images. Further, CNN focuses mainly on the spatial relationship within local receptive fields. However, the process entangles the channel correlation with spatial information. To address this issue, we propose a complex-valued squeeze-excitation network (CV-SENet), where the complex-valued CNN encodes the spatial relationship, and the SENet considers channel-wise important information. Thus, we utilize spatial as well as channel relationships in our work. This, in turn, helps in reducing the speckle noise in the images. The experimental results on several datasets justify the importance of spatial information and inter-channel correlation in classifying PolSAR images.

Keywords: Polarimetric Synthetic Aperture Radar · Convolutional Neural Networks · Complex Value · Squeeze-Excitation · Classification

1 Introduction

Synthetic Aperture Radar (SAR) has become a well-explored remote sensing technique that provides large-scale two-dimensional images with high spatial resolution. These images are mainly that of the Earth's surface reflectivity. This reflectivity is active as the SAR imaging system is a radar system that illuminates the surface with microwave pulses. Radar receives the back-scattered electromagnetic signal from the terrain [4]. The active nature of radar forms the basis of image capturing, which is independent of a light source or an illumination source, making it ideal for day and night imaging. Due to longer wavelengths, the

The work is supported by the Indian Space Research Organisation under Grant No.: ISRO/RES/3/858/19-20.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonopoulos et al. (Eds.): ICPR 2024, LNCS 15302, pp. 270–286, 2025.
https://doi.org/10.1007/978-3-031-78166-7_18

microwave spectral range makes the SAR system capable of capturing images through atmospheric barriers such as clouds, rain, dust, mist, fog, etc. [22].

Data collection and image formation are significant parts of a SAR imaging system. A SAR image is a two-dimensional array of complex-valued pixels containing amplitude and phase information. Polarimetric SAR (PolSAR) is a type of SAR system that uses different polarization channels to characterize targets. These polarization channels are horizontal transmitting and horizontal receiving (HH), horizontal transmitting and vertical receiving (HV), vertical transmitting and horizontal receiving (VH), and vertical transmitting and vertical receiving (VV). Any PolSAR system operates in one of the three modes: Single polarization (HH or VV), Dual polarization (HH and HV or VH and VV), and Quad/Full polarization (HH, HV, VH, and VV).

Classification of PolSAR images is critical for land cover analysis. Information about various aspects such as urban planning, agricultural planning, forest mapping after some natural calamity [27], mapping of glacier melting [28], etc., can be effectively obtained from such an analysis. Deep learning-based methods such as convolutional neural networks (CNN) enjoy state-of-the-art accuracies in the image classification task. However, these methods produce sub-optimal results for complex-valued PolSAR images. Further, the CNN exploits the spatial relationship within local receptive fields. It cannot extract the channel-wise dependencies that play an essential role in the classification task. In order to address this issue, we propose a complex-valued squeeze-excitation network (CV-SENet) on the backbone of complex-valued CNN (CV-CNN) [37] and squeeze-excitation network (SENet) [13]. The proposed CV-SENet adaptively selects channel-wise important features by suppressing others. The combination of SENet and CV-CNN focuses on both spatial as well as channel wise important features by reducing speckle noise. The significant features of our paper can be summarized as:

- We propose a complex-valued squeeze and excitation network (CV-SENet) that combines CV-CNN and SE-Net to extract spatial as well as channel-wise important features for PolSAR image classification.
- Proposed light weight network architecture is robust to speckle noise.
- The network is evaluated on diverse sets of data.

2 Related Work

PolSAR provides terrain information under day and night conditions and all weather conditions. Considering the nature of the data, many unsupervised statistical approaches have been proposed for the classification task. For extracting the polarimetric characteristics of the Entropy-Alpha classes, analysis of the polarimetric signatures using H/α decomposition has been examined [25]. For fitting simple back-scatterer mechanisms to polarimetric SAR data, a scattering model employing three components has been proposed where single bounce,

double bounce, and volume scattering mechanisms have been explored [11]. Various Eigenvector-Eigenvalue-based decomposition techniques are used, and the classification is then done using a Wishart-supervised classifier.

Since the coherency and covariance matrix of PolSAR data follows Wishart distribution, the Wishart mixture model has also been implemented for terrain classification where the model parameters are estimated using the Expectation-Maximization algorithm [5]. When it comes to pattern recognition tasks, classification performance hugely depends on the feature extraction part. These extracted features include back-scattering elements [6], target decomposition-based features [2], and other statistic features [8]. Statistical methods designed to extract such features are usually class-specific and involve manual trial and error. These methods may provide better performance, but these methods require deep domain knowledge, and also, knowledge about specific radar configurations is required for performing feature extraction [3]. This is where multistage deep learning models have scope for automation [39].

CNNs have demonstrated superior performance with their hierarchical feature extraction capabilities [1]. The fundamental task of ship detection for SAR-based surveillance has been carried out using CNNs [7] on Sentinel-1 data. Furthermore, automatic target recognition for SAR has also been efficiently implemented by using a combination of CNNs with support vector machines [30]. In PolSAR images, the data is in the form of complex-valued pixels, and conventional CNNs accept only real numbers. Usually, a polarimetric coherency matrix (T_3) or a polarimetric covariance matrix (C_3) is used for the image classification task. These matrices contain six channels of complex-valued arrays. Using T_3 matrix, real-valued CNN has been implemented where a new 6-D real vector representation has been specifically tailored for the classification of PolSAR data [38]. The edges can play an important role in data representation. Hence, edge-aware methods [12,24] perform quite well for classification. Dual branch fully convolutional neural network method [12] has considered edge-related features for classification purposes. In one of the works, the Haar wavelet-based feature has been incorporated in CNN framework [17].

In some works, the real part and imaginary part of the data are fed separately as a 9-D feature vector [37]. A 6-D complex feature vector is also fed into Complex-Valued CNN, where the entire CNN architecture handles complex numbers [37,38]. Experiments are also performed with only the real-valued coherency matrix elements (T_{11} , T_{22} , T_{33}), where a newly tailored 6-D feature vector representation has been proposed [31]. Backpropagation in the complex-valued CNN domain has been implemented using Complex Generalized Derivative and Complex Conjugate Derivative, and it has shown improvement over real-valued MLPs [15].

The small image patch often contains a single class, whereas large patches consist of multiple classes. Hence, complex-valued multi-scale CNN [36] has been proposed to learn local to global information. However, the method does not explicitly consider the significance of different channels. A work similar to complex-valued CNN and multiple-scale has been proposed with residual

attention enhanced U-Net [26]. The UNet reduces the semantic gap between the encoder and the decoder, along with pixel and channel attention mechanisms. The importance of channel-wise information has been emphasized in [34], where the PolSAR image has been decomposed into three channels according to the scattering mechanism before sending them to CNN channels. Each channel's output is fused together and fed into fully connected layers for classification. The correlation among different feature maps has been explored by considering long short-term memory (LSTM) in complex-valued convolutional network [10]. The LSTM works well for sequential data, but the PolSAR data are generally represented by covariance or coherence matrix. The elements of these matrices may not be sequential in nature. Moreover, training an LSTM network requires a lot of data, which is seldom available for PolSAR data. In our work, we propose a complex-valued squeeze and excitation network to deal with channel-wise significant information along with spatial information.

3 Proposed Method

First we discuss about the representation of the data followed by description of the architecture.

3.1 PolSAR Data Representation

Horizontally polarized wave can create a back-scattered wave with horizontal or vertical polarization. Similarly, vertical wave can also produce both kind of polarization. This set of backscattering properties is contained in 2×2 Sinclair matrix as

$$S = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix}. \quad (1)$$

Here, scattering element S_{HH} signifies horizontal transmission and horizontal reception of polarization channel. The other three terms in S are to be interpreted in the same manner. For monostatic back-scattering, $S_{HV} = S_{VH}$ [5]. Hence, S for our case becomes the associated target vector Ω , that can be written as

$$\Omega = \frac{1}{\sqrt{2}} [S_{HH} \sqrt{2}S_{HV} S_{VV}]^T \quad (2)$$

Using Eq. 2, the 3×3 Covariance matrix C_3 is obtained as

$$C_3 = \langle \Omega \cdot \Omega^{*T} \rangle = \left\langle \begin{bmatrix} |\Omega_1|^2 & \Omega_1 \Omega_2^* & \Omega_1 \Omega_3^* \\ \Omega_2 \Omega_1^* & |\Omega_2|^2 & \Omega_2 \Omega_3^* \\ \Omega_3 \Omega_1^* & \Omega_3 \Omega_2^* & |\Omega_3|^2 \end{bmatrix} \right\rangle \quad (3)$$

Here, $\langle \cdot \rangle$ represents spatial averaging operation that forms multi-looking data. Putting values of Eq. 2 in Eq. 3,

$$C_3 = \begin{bmatrix} \langle S_{HH} S_{HH}^* \rangle & \langle S_{HH} S_{HV}^* \rangle & \langle S_{HH} S_{VV}^* \rangle \\ \langle S_{HV} S_{HH}^* \rangle & \langle S_{HV} S_{HV}^* \rangle & \langle S_{HV} S_{VV}^* \rangle \\ \langle S_{VV} S_{HH}^* \rangle & \langle S_{VV} S_{HV}^* \rangle & \langle S_{VV} S_{VV}^* \rangle \end{bmatrix}. \quad (4)$$

3.2 Network Architecture

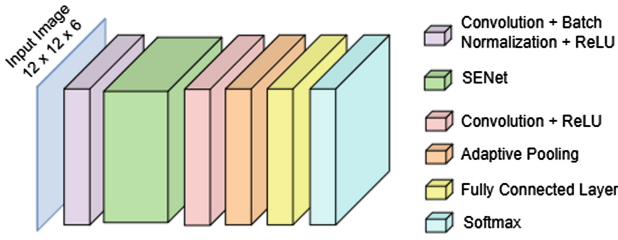


Fig. 1. Network Architecture

The input PolSAR image is first divided into 12×12 patches. We compute the covariance matrix C_3 from these patches. The C_3 matrix of Eq. 3 or 4 is Hermitian. The off-diagonal elements are complex-valued, which are conjugate symmetric about the main diagonal real-valued elements. Thus the six elements of the upper triangle of C_3 matrix are sufficient for representation. The data in C_3 matrix is distributed over 6 channels as follows,

$$C_3 = (C_{11}, C_{12}, C_{13}, C_{22}, C_{23}, C_{33}) \quad (5)$$

The C_3 matrices are passed through the proposed architecture (Shown in Fig. 1) based on complex-valued SE-Net (CV-SENet). The CV-SENet is developed based on CV-CNN layers with complex-valued CV-SE block. Moreover, batch normalization, complex ReLU, pooling, fully connected layers are used along with softmax for classification.

Complex-Valued Convolution. The complex-valued convolution takes care of the complex values of the PolSAR image. The complex-valued feature output of l^{th} layer $M_k^{(l)} \in C^{F \times F \times K \times I}$ is convolved with complex-valued kernel $w_{ik}^{(l+1)} \in C^{B_1 \times H_1 \times K}$ along with an addition of bias to produce a feature $F_i^{(l+1)}$ at $(l+1)^{th}$ layer.

$$\begin{aligned} F_i^{(l+1)} &= \sum_{k=1}^K w_{ik}^{(l+1)} * M_k^{(l)} + b_i^{(l+1)} \\ &= \sum_{k=1}^K (\Re\{w_{ik}^{(l+1)}\} \cdot \Re\{M_k^{(l)}\}) - \Im\{w_{ik}^{(l+1)}\} \cdot \Im\{M_k^{(l)}\} \\ &\quad + j \sum_{k=1}^K (\Re\{w_{ik}^{(l+1)}\} \cdot \Im\{M_k^{(l)}\}) + \Im\{w_{ik}^{(l+1)}\} \cdot \Re\{M_k^{(l)}\} \\ &\quad + b_i^{l+1} \end{aligned} \quad (6)$$

The produced feature is passed through complex-valued batch normalization, followed by non-linear activation.

Complex-Valued Batch Normalization with Complex ReLU. The main steps of batch normalization [16] is subtraction by the expected values and division by standard deviation. Two associated learnable parameters α and β are present to scale and shift the feature.

$$\begin{aligned}\hat{F}_i^{(l+1)} = & \Re\{\alpha\} \frac{\Re\{F_i^{(l+1)}\} - E[\Re\{F_i^{(l+1)}\}]}{\sqrt{\text{Var}(\Re\{F_i^{(l+1)}\}) + \epsilon}} + \Re\{\beta\} \\ & + \Im\{\alpha\} \frac{\Im\{F_i^{(l+1)}\} - E[\Im\{F_i^{(l+1)}\}]}{\sqrt{\text{Var}(\Im\{F_i^{(l+1)}\}) + \epsilon}} + \Im\{\beta\},\end{aligned}\quad (7)$$

Scaling and shifting the features helps in accelerating the training process. The reason is that the method eliminates the issues of change in feature distribution among the layers by enforcing β mean and α variance. The normalized features with controlled mean and variance behave better in subsequent layers for faster convergence [29].

The batch-normalized feature are passed through complex-ReLU activation unit.

$$M_i^{l+1} = \text{ReLU}(\Re\{F_i^{l+1}\}) + j\text{ReLU}(\Im\{F_i^{l+1}\}), \quad (8)$$

where $\text{ReLU}(x) = \max(0, x)$ is the non-linear excitation. M_i^{l+1} is passed through squeeze-excitation network to produce a feature that is enriched with inter-channel correlation.

Complex-Valued Squeeze-Excitation Network. The main component of Complex-valued Squeeze-Excitation Network (CV-SENet) is the complex SE block [13]. The input to our CV-SENet is a volume consisting of different channels, where each channel represents each element of the C_3 matrix as discussed in Eq. 5. The diagonal elements C_{11}, C_{22}, C_{33} depict the power of different polarization channels, whereas the off-diagonal elements C_{12}, C_{13}, C_{23} represent the correlation between different polarization channel. SE-Net can model the channel inter-correlation to improve convolutional features. Hence, SE-Net is quite suitable for exploring the inter-dependencies among the different polarization channels of Pol-SAR data. As a result, SE-Net assists in effectively combining polarimetric-domain features with spatial-domain features. It enables significant information propagation towards the next layer. SE block performs squeeze and excitation operation. We extend the conventional SE block to deal with complex values. The squeeze operation is performed by

$$\mathcal{F}_{sq}(M_i) = \frac{1}{n \times d} \sum_{x=1}^n \sum_{y=1}^d (\Re\{M_i(x, y)\} + j\Im\{M_i(x, y)\}). \quad (9)$$

This is equivalent to average pooling per channel, which represents the channel-wise statistics. The dependencies among the channel is exploited by excitation operation by a gating mechanism with sigmoid activation as

$$e_i = \mathcal{F}_{ex}(s_i) = \frac{1}{1 + e^{-\Re\{W'_1 \tilde{s}_i\}}} + j \frac{1}{1 + e^{-\Im\{W'_1 \tilde{s}_i\}}} \quad (10)$$

where $s_i = \mathcal{F}_{sq}(M_i)$, $\tilde{s}_i = \max(W_1 s_i, 0)$. e_i is then multiplied channel-wise with M_i to produce the output feature o_i .

The advantage of using SENet for PolSAR images is that it captures channel-wise dependencies that are learned by extracting non-mutually exclusive and nonlinear relationships between the individual channels. This is achieved by the excitation module, which follows the squeeze model. The main advantage of using a squeeze network is to interpret the local descriptors that capture the statistics for the whole image. Another reason for exploring the efficiency of SE blocks for SAR image classification is that the speckle noise is spread across channels. The SE block helps reduce noise by suppressing unwanted information.

Pooling and Fully Connected Layer. We use adaptive average pooling, where an average value of 2×2 window is computed in the first iteration. However, the window size and stride gets updated according to the performance of the network. After pooling, the feature map $M_k^{(l)}$ goes through the fully connected layer as

$$M_i^{(l+1)} = f_c \left(\Re\{F_i^{(l+1)}\} \right) + j f_c \left(\Im\{F_i^{(l+1)}\} \right), \quad (11)$$

where $F_i^{l+1} = \sum_{k=1}^K w_{ik}^{(l+1)} \cdot M_k^{(l)} + b_i^{l+1}$.

Loss Function. We use the categorical cross-entropy loss for optimizing the model. The loss function is given by

$$L = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^C y_{i,j} \log(\sigma(M_{i,j})) \quad (12)$$

where $y_{i,j} = 1$ if M_i belongs to class j , otherwise $y_{i,j} = 0$. i indices the training sample, whereas j denotes the class. $\sigma(M_{i,j})$ is the output of the softmax function. Hence, $\sigma(M_{i,j})$ denotes the probability of i^{th} sample belongs to the j^{th} class.

$$\sigma(M_{i,j}) = \frac{\exp\{M_{i,j}\}}{\sum_{k=1}^C \exp\{M_{i,k}\}} \quad (13)$$

4 Experimental Results

4.1 Datasets and Experimental Setting

We consider four datasets of different geographic regions to demonstrate the effectiveness of our architecture. Out of four datasets, three are publicly available [9], namely Flevoland15, Flevoland7, and Landes. The fourth dataset is

Table 1. Datasets Details

Sl. No.	Name	Region	Sensor	Date	Resolution	Class
1	Flevoland15	Flevoland, Netherlands	AIRSAR	16-Aug-1989	750×1024	15
2	Flevoland7	Flevoland, Netherlands	AIRSAR	16-Jun-1991	750×700	7
3	Landes	Landes, France	AIRSAR	19-Jun-1991	1050×1000	6
4	Mysore	Mysore, India	RADAR SAT-2	29-Aug-2017	3489×3352	11

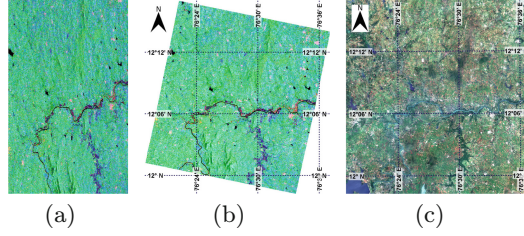


Fig. 2. Mysore dataset: (a) Pseudo-color Freeman-Durden RGB image in slant-range resolution, (b) Pseudo-color Freeman-Durden RGB image in ground-range resolution (c) Sentinel-2 optical image

provided by the Space Application Centre-Indian Space Research Organization (SAC-ISRO). The details of the datasets are given in Table 1. All the datasets are full-polarimetric L-band data. Flevoland15 and Flevoland7 datasets are constructed with the polSAR data of agricultural land in the Flevoland region containing different types of crops. The Landes dataset is of pine forests in the Landes region. It includes trees of different ages. The Mysore dataset (shown in Fig. 2) is captured from the area between 11.97 – 12.284 N latitude and 76.371 – 76.561 E longitude. The original dataset is in the slant-range resolution. The data is pre-processed by Range-Doppler terrain correction using the SNAP tool. The tool helps us to convert images in ground-range resolution, to remove the topographic effect, and to rectify the geoid. The ground truth mask is manually constructed with the help of ESRI shapefile containing the placemark points of different crop fields. The shapefile is constructed at the same time when the data was originally captured using a hand-held GPS receiver. Furthermore, we select Sentinel-2 optical image of the nearest duration having low cloud coverage. The placemark points are inlaid on the optical image, and the ground truth polygon mask is built using the QGIS tool.

The entire image of the dataset is divided into 12×12 patches. All the experiments are carried out using the PyTorch framework. The datasets are divided into training and testing data, where 30% patches are chosen for testing. Tesla P100-PCIE with 16GB graphics RAM is used to train the models. For training, the Adam optimizer is used on the categorical cross-entropy loss.

4.2 Data Augmentation for Training

The datasets are heterogeneous in nature. Different classes of a dataset have a different number of training patches. A few classes have very few classes for

training, so those classes are misclassified into other classes. Hence, we perform data augmentation during training. For this purpose, we create a training set using the help of different speckle-noise filters, namely Boxcar [23], Lee refined [21] and Mean shift [20]. Hence, the original unfiltered training data is augmented with the output of the mentioned filters.

Boxcar filter considers a square window of fixed size and replaces the center pixel with the average number of pixels within the window. The nature of the boxcar filter is such that it disregards the edges. Hence, it works quite well for homogeneous regions but produces sub-optimal results for heterogeneous regions. The Lee-refined filter behaves better for edges because it considers only similar pixels for averaging within edge-aligned windows. In comparison, the mean-shift method considers a moving window instead of a fixed static window. It considers both spatial and spectral information along with weighted averaging of pixels. Mean-shift works quite well to reduce speckle noise. Thus, the selected filters complement each other in terms of removing noise and preserving edge details.

4.3 Results

Here, we show the experimental results for different datasets. We compare our results with some classical and deep learning-based methods. Since the covariance matrix of the homogeneous PolSAR data follows the Wishart distribution and the heterogeneous data can be modeled better using Wishart mixture models, we first compare our results with the methods based on these distributions. Our model has two variations based on the training data: CV-SENet is trained without the data-augmentation method, and CV-SENet+ is boosted with augmented data. Further, the classical techniques are trained and tested with speckle-filtered data, whereas our CV-SENet is trained and tested with the raw data without any filtering.

Table 2 shows the comparison between the results obtained using classical methods (columns 2–5) and our model. One can observe that our CV-SENet produces better or similar results as compared to the classical methods. The overall accuracy (OA) for WMM-CRF is the same as CV-SENet+. Hence, one may doubt the requirements of our model. Note that CV-SENet takes unfiltered data as input, whereas the WMM and its different versions take filtered data as input. For the unfiltered data, the classical techniques produce sub-optimal results. The accuracies of the Wishart and the WMM classifiers go down to 46.58% and 50.04%, respectively. Considering these accuracies for unfiltered data, our CV-SENet improves the results by a large margin. This is because the employed convolution layers and the SE block suppress the noise across different channels. Hence, the effect of noise is relatively less for our classification results. Another important point to observe from the table is that for some classes, the performance of the proposed CV-SENet is a bit inferior as compared to the other methods. This is because those classes have very few patches for training. For example, the *Building* class has only two patches for training, whereas the *Wheat3* has 116 patches. This makes the training difficult for CV-SENet. Here,

Table 2. Classification results of Flevoland15 dataset

Class Label	Wishart [5]	WMM [5]	WMM-MRF [24]	WMM-CRF [24]	CV-SENet	CV-SENet+
Water	64.77	99.21	99.99	99.33	100	99.21
Forest	90.05	91.95	99.99	99.93	100	100
Lucerne	92.28	96.82	99.70	99.37	100	97.56
Grasses	77.00	92.50	99.54	99.26	77.78	98.59
Peas	91.77	97.61	99.77	99.73	100	100
Barley	90.80	97.18	99.94	99.90	66.67	100
BareSoil	48.45	96.66	99.79	97.08	100	100
Beet	92.16	94.73	94.18	98.17	100	100
Wheat2	76.19	87.20	99.78	99.71	100	96.74
Wheat3	91.11	95.45	99.90	99.82	100	99.49
Stembeans	95.11	97.19	97.76	99.64	100	100
Rapeseed	75.28	87.46	98.21	98.96	95.83	100
Wheat	88.58	93.58	99.53	99.74	97.44	99.35
Buildings	89.29	80.80	90.57	91.86	0	100
Potatoes	88.88	90.61	96.17	97.87	100	100
OA	84.56	93.72	98.80	99.18	97.12	99.18

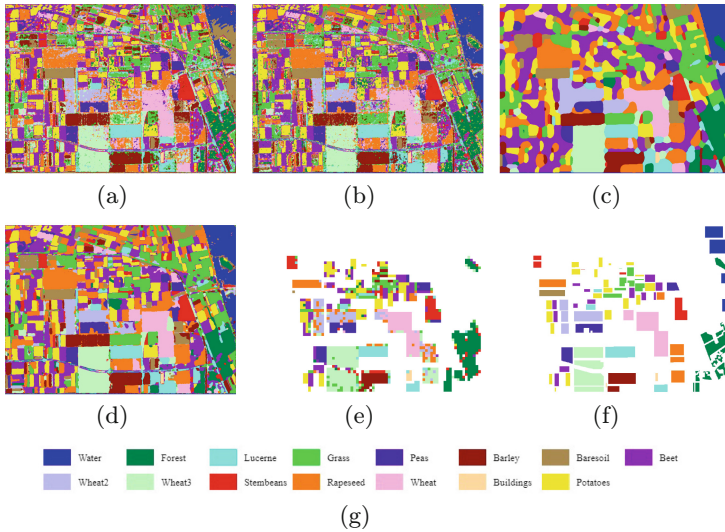


Fig. 3. Results on Flevoland15 (a) Wishart Classifier, (b) WMM Classifier (c) WMM-MRF (d) WMM-CRF, (e) CV-SENet+, (f) Ground truth, and (g) Ground-truth labels

the proposed CV-SENet+ performs quite well as it is trained with augmented data. The quantitative results can be verified visually from Fig. 3.

Table 3 shows the results of our method on Flevoland7 dataset. One can note that the proposed CV-SENet performs better than the existing classical techniques even with unfiltered PolSAR data, whereas the classical techniques

Table 3. Classification results of Flevoland7 dataset

Class Label	Wishart [5]	WMM [5]	WMM-MRF [24]	WMM-CRF [24]	CV-SENet	CV-SENet+
Wheat	95.59	98.98	98.80	98.98	99.12	100
Rapeseed	96.41	99.91	99.94	99.94	100	100
Barley	98.29	99.39	99.11	99.26	100	100
Lucerne	57.53	94.97	99.04	96.25	85.71	100
Potatoes	95.98	99.15	95.54	99.71	98.18	100
Beet	33.00	96.99	94.20	98.63	93.94	97.93
Peas	50.74	92.87	100	93.45	100	100
OA	89.61	98.88	98.33	99.09	98.48	99.81

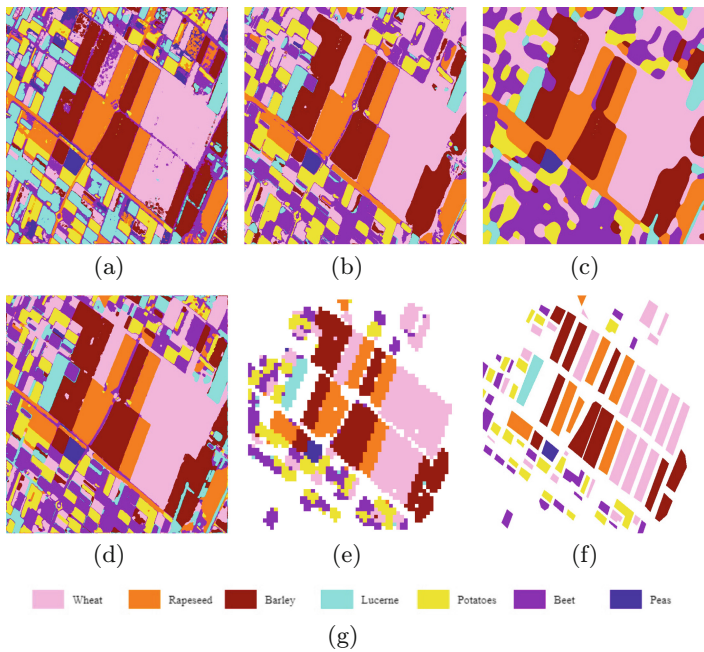


Fig. 4. Results on Flevoland7 (a) Wishart Classifier, (b) WMM Classifier (c) WMM-MRF (d) WMM-CRF, (e) CV-SENet+, (f) Ground truth, and (g) Ground-truth labels

use the speckle-filtered data. The results can be compared visually in Fig. 4. One can observe that our method is able to group pixels quite well as compared to the existing methods.

Table 4 depicts the results and comparisons with other approaches for Landes dataset. One can see that even for difficult classes such as C1 and C6, where the Wishart classifier produces 44.54 and 30% accuracies, our CV-SENet+ classifies the data with 98.33 and 96.39% accuracies. These results are reflected in Fig. 5, where the classification maps are shown.

Table 4. Classification results of Landes dataset

Class Label	Wishart [5]	WMM [5]	WMM-MRF [24]	WMM-CRF [24]	CV-SENet	CV-SENet+
C1	44.54	84.41	95.40	95.54	91.71	98.33
C2	65.71	79.98	95.35	98.32	100	94.20
C3	69.92	97.62	99.94	100	100	100
C4	88.40	97.31	99.85	99.69	99.33	100
C5	56.93	95.41	99.98	100	100	100
C6	30.00	78.17	92.39	91.87	81.16	96.39
OA	66.42	90.73	97.64	97.67	95.40	98.85

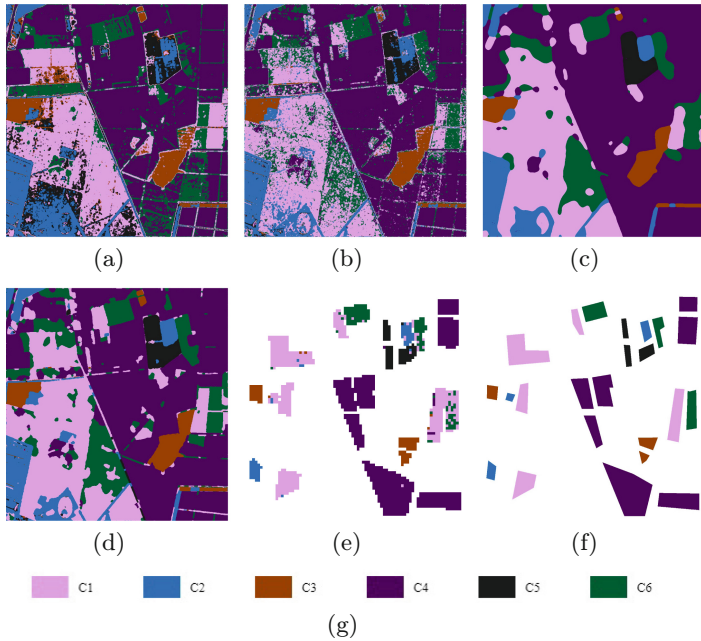


Fig. 5. Results on Landes:(a) Wishart Classifier, (b) WMM Classifier (c) WMM-MRF (d) WMM-CRF, (e) CV-SENet+, (f) Ground truth, and (g) Ground-truth labels

Table 5 contains the results and comparisons with other approaches for Mysore dataset. The dataset is a bit complex as compared to other datasets, as the class regions are scattered all over the image. Moreover, a few classes has a very few patches (as less as 5). Hence, training and testing is difficult for these classes using CV-SENet. Thus, we perform augmentation on the data as discussed earlier, and show the results for CV-SENet+ only. The CV-SENet+ is able to outperform the existing approaches quite well.

Table 6 shows the comparison with existing neural network based methods using the Flevoland15 dataset. Here, the efficiency of the proposed method is compared with thirteen existing neural network based models, including the complex valued-CNN and transformer based models. One can observe that the

Table 5. Classification results of Mysore dataset

Class Label	Wishart [5]	WMM [5]	WMM-MRF [24]	WMM-CRF [24]	CV-SENet+
Ragi	61.98	85.99	95.84	95.73	100
Ginger	58.37	69.35	93.01	92.54	100
Rice	93.00	96.50	98.35	98.13	100
Urban	91.93	96.53	99.51	96.86	100
Water	94.27	96.34	99.68	97.89	100
Arecanut	59.03	75.07	90.47	90.32	100
Banana	28.09	61.25	73.87	66.51	100
Sugarcane	65.78	76.70	81.23	89.05	100
Coconut	59.13	94.13	87.68	89.84	100
Fallow	41.18	51.61	84.74	80.97	100
Magnesite Mine	52.70	82.75	92.23	91.15	81.82
OA	85.64	92.63	97.22	96.77	99.20

Table 6. Performance (OA) comparisons using Flevoland15 dataset

WCAE [35]	SPCNN [18]	Te-CNN [14]	MCFCNN [34]	RV-CNN [37]	CV-CNN [37]	CVMS [36]
93.31	96.90	96.63	95.83	95.30	96.20	97.74
DBFCN [12]	CV-RAU [26]	Conv-LSTM [10]	CV-SANP [19]	MCPT [32]	[33]MHCV	CV-SENet+
98.76	98.83	98.58	95.80	97.00	98.92	99.18

Table 7. Ablation Study (OA)

Experiment				Datasets		
Data Augmentation			SE-Net	Flevoland15	Flevoland7	Landes
Boxcar	Lee refined	Mean shift				
✗	✗	✗	✗	82.42	96.95	94.88
✗	✗	✗	✓	97.12	98.48	95.40
✓	✓	✗	✓	98.17	99.66	98.72
✓	✓	✓	✓	99.18	99.81	98.85

proposed method is able to produce best results as compared to the existing approaches.

4.4 Ablation Study

In order to analyse the effectiveness of the main components of CV-SENet, we perform ablation experiments on three datasets. Here, we study the effectiveness of the SE-Net and Data augmentation techniques through overall accuracy in Table 7. For data augmentation, we consider three techniques namely Boxcar [23], Lee refined [21] and Mean shift [20]. First, we check the results without proposed SE-Net and data-augmentation (first row of the results). Here, we use

CV-CNN architecture of Fig. 1 without SE-Net. One can observe that the results are inferior as compared to the proposed method (the last row). The results have improved when we employ proposed SE-Net into the architecture (the second row) as compared to the first experiment. When the data-augmentation techniques are used along with the CV-SENet, the results are even better. Hence, the combination of data augmentation (Boxcar, Lee refined, and Mean Shift) and the CV-SENet produces the best results.

5 Conclusion

In this paper, we proposed a complex-valued SENet architecture to classify land covers from PolSAR images. In the classification task, channel-wise important information plays an essential role along with its spatial content. Convolutional layers take care of the spatial correlation, whereas the inter-channel dependencies are exploited by squeeze and excitation network (SE-Net). The proposed models have produced promising results on unfiltered data, confirming that the squeeze-excitation, when configured with complex-valued deep learning methods appropriately, can suppress the effect of speckle noise while classifying the PolSAR image. The method is shown to perform exceptionally well on the four datasets.

References

1. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6 (2017). <https://doi.org/10.1109/ICEngTechnol.2017.8308186>
2. Arii, M., van Zyl, J.J., Kim, Y.: Adaptive model-based decomposition of polarimetric SAR covariance matrices. *IEEE Trans. Geosci. Remote Sens.* **49**(3), 1104–1113 (2011). <https://doi.org/10.1109/TGRS.2010.2076285>
3. Cao, Y., Wu, Y., Zhang, P., Liang, W., Li, M.: Pixel-wise PolSAR image classification via a novel complex-valued deep fully convolutional network (2019)
4. Chaturvedi, S.K.: Study of synthetic aperture radar and automatic identification system for ship target detection. *J. Ocean Eng. Sci.* **4**(2), 173–182 (2019). <https://doi.org/10.1016/j.joes.2019.04.002>
5. Chaudhari, N., Mitra, S.K., Chirakkal, S., Mandal, S., Putrevu, D., Misra, A.: Discrimination of multi-crop scenarios with polarimetric SAR data using Wishart mixture model. *J. Appl. Remote Sens.* **15**(3), 1–21 (2021). <https://doi.org/10.1117/1.JRS.15.034514>
6. Chen, C.T., Chen, K.S., Lee, J.S.: The use of fully polarimetric information for the fuzzy neural classification of SAR images. *IEEE Trans. Geosci. Remote Sens.* **41**(9), 2089–2100 (2003). <https://doi.org/10.1109/TGRS.2003.813494>
7. Cozzolino, D., Di Martino, G., Poggi, G., Verdoliva, L.: A fully convolutional neural network for low-complexity single-stage ship detection in sentinel-1 SAR images. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 886–889 (2017). <https://doi.org/10.1109/IGARSS.2017.8127094>

8. Dabboor, M., Collins, M.J., Karathanassi, V., Braun, A.: An unsupervised classification approach for polarimetric SAR data based on the chernoff distance for complex wishart distribution. *IEEE Trans. Geosci. Remote Sens.* **51**(7), 4200–4213 (2013). <https://doi.org/10.1109/TGRS.2012.2227755>
9. ESA: Earth online - ESA. <https://step.esa.int/main/toolboxes/polsarpro-v6-0-biomass-edition-toolbox/>
10. Fang, Z., Zhang, G., Dai, Q., Xue, B.: PoLSAR image classification based on complex-valued convolutional long short-term memory network. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022). <https://doi.org/10.1109/LGRS.2022.3146928>
11. Freeman, A., Durden, S.: A three-component scattering model for polarimetric SAR data. *IEEE Trans. Geosci. Remote Sens.* **36**(3), 963–973 (1998). <https://doi.org/10.1109/36.673687>
12. Gao, F., et al.: Polarimetric SAR image classification based on edge-aware dual branch fully convolutional network. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 4728–4731 (2021). <https://doi.org/10.1109/IGARSS47720.2021.9554518>
13. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018). <https://doi.org/10.1109/CVPR.2018.00745>
14. Hua, W., Xie, W., Jin, X.: Three-channel convolutional neural network for polarimetric SAR images classification. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **13**, 4895–4907 (2020). <https://doi.org/10.1109/JSTARS.2020.3018161>
15. Hänsch, R., Hellwich, O.: Classification of polarimetric SAR data by complex valued neural networks (2009)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, pp. 448–456. ICML'15, JMLR.org (2015). <https://doi.org/10.5555/3045118.3045167>
17. Jamali, A., Mahdianpari, M., Mohammadimanesh, F., Bhattacharya, A., Homayouni, S.: PoLSAR image classification based on deep convolutional neural networks using wavelet transformation. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022). <https://doi.org/10.1109/LGRS.2022.3185118>
18. Jiao, C., et al.: Self-paced convolutional neural network for PoLSAR images classification. *Remote Sens.* **11**(4), 424 (2019). <https://doi.org/10.3390/rs11040424>, <https://www.mdpi.com/2072-4292/11/4/424>
19. Kuang, Z., Bi, H., Li, F.: Complex-valued self-supervised PoLSAR image classification integrating attention mechanism. In: IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium, pp. 5958–5961 (2023). <https://doi.org/10.1109/IGARSS52108.2023.10281737>
20. Lang, F., Yang, J., Li, D., Shi, L., Wei, J.: Mean-shift-based speckle filtering of polarimetric SAR data. *IEEE Trans. Geosci. Remote Sens.* **52**(7), 4440–4454 (2014). <https://doi.org/10.1109/TGRS.2013.2282036>
21. Lee, J.S., Grunes, M., de Grandi, G.: Polarimetric SAR speckle filtering and its implication for classification. *IEEE Trans. Geosci. Remote Sens.* **37**(5), 2363–2373 (1999). <https://doi.org/10.1109/36.789635>
22. Lee, J.S., Pottier, E.: Polarimetric Radar Imaging: From Basics to Applications (2009). <https://doi.org/10.1201/9781420054989>
23. Lee, J.S., Pottier, E.: Polarimetric radar imaging: from basics to applications. CRC press (2017)

24. Nilam Chaudhari, Suman K. Mitra, S.M.S.C.D.P., Misra, A.: Edge-preserving classification of polarimetric SAR images using wishart distribution and conditional random field. *Int. J. Remote Sens.* **43**(6), 2134–2155 (2022). <https://doi.org/10.1080/01431161.2022.2054293>
25. Ouarzeddine, M., Souissi, B., Belhadj-Aissa, A.: Target detection and characterization using h/α decomposition and polarimetric signatures. In: 2006 2nd International Conference on Information Communication Technologies. vol. 1, pp. 395–400 (2006). <https://doi.org/10.1109/ICTTA.2006.1684402>
26. Ren, S., Zhou, F.: PoLSAR image classification with complex-valued residual attention enhanced U-Net. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 3045–3048 (2021). <https://doi.org/10.1109/IGARSS47720.2021.9554303>
27. Saini, O., Bhardwaj, A., Chatterjee, R.: Mapping of Sundarbans mangrove forest with ALOS PaLSAR dual polarimetric SAR data using SVM classifier (2019)
28. Scher, C., Steiner, N., McDonald, K.: Mapping seasonal glacier melt across the Hindu Kush Himalaya with time series SAR (2020). <https://doi.org/10.5194/tc-2020-181>
29. Shah, B., Bhatt, K., Mandal, S., Mitra, S.K.: Emotioncaps - facial emotion recognition using capsules. In: Yang, H., Pasupa, K., Leung, A.C.S., Kwok, J.T., Chan, J.H., King, I. (eds.) *Neural Information Processing*, pp. 394–401. Springer International Publishing, Cham (2020)
30. Wagner, S.A.: SAR ATR by a combination of convolutional neural network and support vector machines. *IEEE Trans. Aerosp. Electron. Syst.* **52**(6), 2861–2872 (2016). <https://doi.org/10.1109/TAES.2016.160061>
31. Wang, H., Xu, F., Jin, Y.Q.: A review of PoLSAR image classification: from polarimetry to deep learning. In: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 3189–3192 (2019). <https://doi.org/10.1109/IGARSS.2019.8899902>
32. Wang, W., Wang, J., Lu, B., Liu, B., Zhang, Y., Wang, C.: MCPT: mixed convolutional parallel transformer for polarimetric SAR image classification. *Remote Sens.* **15**(11), 2936 (2023). <https://doi.org/10.3390/rs15112936>, <https://www.mdpi.com/2072-4292/15/11/2936>
33. Wang, W., Wang, J., Quan, D., Yang, M., Sun, J., Lu, B.: PoLSAR image classification via a multigranularity hybrid CNN-ViT model with external tokens and cross-attention. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **17**, 8003–8019 (2024). <https://doi.org/10.1109/JSTARS.2024.3384420>
34. Wang, Y., Cheng, J., Zhou, Y., Zhang, F., Yin, Q.: A multichannel fusion convolutional neural network based on scattering mechanism for PoLSAR image classification. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022). <https://doi.org/10.1109/LGRS.2020.3047635>
35. Xie, W., et al.: PoLSAR image classification via wishart-AE model or wishart-CAE model. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **10**(8), 3604–3615 (2017). <https://doi.org/10.1109/JSTARS.2017.2698076>
36. Zhang, L., Zhang, S., Dong, H., Lu, D.: PoLSAR image classification via complex-valued multi-scale convolutional neural network. In: IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, pp. 200–203 (2020). <https://doi.org/10.1109/IGARSS39084.2020.9323621>
37. Zhang, Z., Wang, H., Xu, F., Jin, Y.Q.: Complex-valued convolutional neural network and its application in polarimetric SAR image classification. *IEEE Trans. Geosci. Remote Sens.* **55**(12), 7177–7188 (2017). <https://doi.org/10.1109/TGRS.2017.2743222>

38. Zhou, Y., Wang, H., Xu, F., Jin, Y.Q.: Polarimetric SAR image classification using deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **13**(12), 1935–1939 (2016). <https://doi.org/10.1109/LGRS.2016.2618840>
39. Zhu, X., et al.: Deep learning meets SAR: concepts, models, pitfalls, and perspectives. *IEEE Geosci. Remote Sens. Mag.* **9**(4), 143–172 (2021). <https://doi.org/10.1109/MGRS.2020.3046356>



Probabilistic Fusion Framework Combining CNNs and Graphical Models for Multiresolution Satellite and UAV Image Classification

Martina Pastorino^{1,2(✉)}, Gabriele Moser¹, Fabien Guerra³,
Sebastiano B. Serpico¹, and Josiane Zerubia²

¹ DITEN Department, University of Genoa, Genoa, Italy
martina.pastorino@edu.unige.it

² Inria, Université Côte d'Azur, Sophia-Antipolis, France

³ INRAE, RECOVER, Aix-Marseille University, Aix-en-Provence, France

Abstract. Image classification - or semantic segmentation - from input multiresolution imagery is a demanding task. In particular, when dealing with images of the same scene collected at the same time by very different acquisition systems, for example multispectral sensors onboard satellites and unmanned aerial vehicles (UAVs), the difference between the involved spatial resolutions can be very large and multiresolution information fusion is particularly challenging. This work proposes two novel multiresolution fusion approaches, based on deep convolutional networks, Bayesian modeling, and probabilistic graphical models, addressing the challenging case of input imagery with very diverse spatial resolutions. The first method aims to fuse the multimodal multiresolution imagery via a posterior probability decision fusion framework, after computing posteriors on the multiresolution data separately through deep neural networks or decision tree ensembles. The optimization of the parameters of the model is fully automated by also developing an approximate formulation of the expectation maximization (EM) algorithm. The second method aims to perform the fusion of the multimodal multiresolution information through a pyramidal tree structure, where the imagery can be inserted, modeled, and analyzed at its native resolutions. The application is to the semantic segmentation of areas affected by wildfires for burnt area mapping and management. The experimental validation is conducted with UAV and satellite data of the area of Marseille, France. The code is available at https://github.com/Ayana-Inria/BAS_UAV_satellite_fusion.

Keywords: graphical models · deep learning · probabilistic fusion · multiresolution imagery · semantic segmentation · wildfires · UAVs

1 Introduction

In the framework of pattern recognition, a semantic segmentation problem, whose goal is to assign a class label to each individual pixel in an image, can

be formalized as a supervised image-classification problem [33]. Within semantic segmentation tasks, the use of multimodal data has been shown to favor accuracy and spatial precision of the classification results [10]. From a computer vision perspective, the development of processing methods that can benefit from multimodal information (e.g., synoptic and detailed views from multiresolution data, different band information from multisensor imagery) and take advantage of the complementary information therein contained presents huge potentials.

Thanks to the advent of deep learning, the performances of semantic segmentation algorithms have significantly improved. However, there are still some challenges. For example, when dealing with remote sensing images, one of the main issues is the variability of features within the same category in the image, leading to confusion in segmentation. Moreover, the availability of training data is a key requirement for deep learning architectures, not always feasible for computer vision applications related to remote sensing. To address these challenges, one way is to leverage contextual and multiscale information for accurate segmentation [29, 30].

For example, focusing on land-cover mapping applications, on the one hand there are satellite imaging sensors, which provide an efficient and large-scale coverage of the Earth surface, thanks to their wide range and short revisit time. Optical satellite imagery with spatial resolution as fine as 10 m is made available by space missions with open data policies (e.g., the ESA Copernicus program). However, optical satellite sensors are sensitive to weather conditions and Sun illumination. On the other hand, in recent years, unmanned aerial vehicles (UAV) – or drones – have also sparked a lot of interest thanks to their high flexibility, low-cost, and ability to cover wide areas during the day or night [43]. UAV monitoring is undertaken at low-to-medium altitudes, thus effectively avoiding the cloud interference, and allowing for very high spatial resolution up to few centimeters. However, the imagery captured by UAVs is typically characterized by a small area coverage, irregular contours, susceptibility to forest cover, making land-cover mapping from UAV imagery a challenging task [1].

The joint availability of satellite and UAV acquisitions of the same geographical zones, with their complementary features, presents a huge potential for semantic segmentation applications and, simultaneously, a big challenge for the development of a method capable to fully take advantage of this multimodal information. The resulting multiresolution fusion task is quite extreme, and currently under-exploited, since the resolution ratio between the input image sources is of the order of the hundreds – a situation that is normally not addressed by traditional multiresolution schemes [6, 13, 29–32, 35–39, 42].

In this paper, two approaches based on deep learning, Bayesian fusion, and probabilistic graphical fusion are proposed for the semantic segmentation of multiresolution imagery with a huge ratio between resolutions. The focus is on binary classification problems, which have many applications in natural disaster management, such as the detection of areas affected by floods, wildfires, or earthquakes [24], the mapping of urban areas and human settlements [8], of snow covers [22], and cloud masking [23]. The first method proposes a pixelwise probabilistic fusion of the multiresolution information after computing the posterior probabilities with separate classifiers – neural networks and decision tree

ensembles – on the multimodal images separately. The parameters of the method are automatically optimized by developing a case-specific formulation of the EM algorithm, based on a pseudo-likelihood-type approximation. The second considers multiresolution fusion in a pyramidal tree graph topology through the marginal posterior mode (MPM) criterion, an extension to the case of great spatial resolution ratio of the approach proposed in [29,30] for multimodal and multiresolution images.

The main novel contributions of this paper are twofold: (i) the development of two semantic segmentation methods for input multiresolution imagery with great mismatch in spatial resolution; (ii) the combination, within the two novel methods, of deep learning, stochastic modeling, decision fusion, and an EM-based automatic parameter optimization.

2 Related Work

Here, we briefly review the literature on semantic segmentation from input multiresolution imagery. Models for multimodal data, in particular multiscale and multiresolution methods, are gaining importance in order to face the requirements of several applications, for example remote sensing [10] and medical image processing [32,37]. The idea is to jointly use multiple images associated with distinct spatial resolutions to benefit from their complementary perspectives.

Wavelet-based methods [25] are often employed to perform multiresolution image processing. In [13] an image segmentation method for human face detection based on multiresolution wavelet transforms and watershed segmentation algorithm is presented. In [6] a wavelet-based multiresolution pyramid applied to multitemporal or multisensor satellite data is combined with a stochastic gradient based on two similarity measures, correlation and mutual information. In [42] several wavelet pyramids aimed at performing invariant feature extraction and accelerating image fusion through multiple spatial resolutions are evaluated.

Deep learning methods are state-of-the-art techniques for computer vision tasks [26]. Fully convolutional networks (FCNs) structurally involve several multiscale processing stages, through their encoder-decoder architecture and their convolutional and pooling layers. In [37], a semantic segmentation model for histopathology whole-slide images, which combines multiresolution context and details via multiple branches of encoder-decoder neural networks, is proposed. A multi-scale representation learning network integrating CNNs and Transformers was proposed in [20] to exploit multi-scale local detailed feature and global contextual information for the segmentation of lesions in lung CT images. The joint potential of CNNs and Transformers for the analysis of local and multiscale information was explored in [35], as well, for the semantic segmentation of urban remote sensing images.

The literature on multiresolution fusion in remote sensing is vast and dates back a few decades [33], with approaches rooted in several methodological areas, such as statistical pattern recognition [15,31], neural networks [17,39], decision fusion [40], kernel-based approaches [36], and Markov random fields [3,38].

Furthermore, with the advent and diffusion of UAV platforms, images with extremely high spatial resolution have become available at a relatively low cost [19]. UAVs are often equipped with simple, lightweight sensors, such as RGB cameras [36] that capture small portions of land. In [36] classification of a high spatial resolution RGB image and a lower spatial resolution hyperspectral image of the same scene is addressed. Contextual information is obtained from the RGB image through color attribute profiles, and spectral information is extracted from the hyperspectral image; a composite decision fusion strategy exploiting kernel-based techniques is proposed.

3 Methodology

The aim of the proposed techniques is to perform the fusion of multiresolution imagery – with big mismatch in spatial resolution – for binary semantic segmentation tasks without the need of resampling techniques. The two proposed approaches integrate stochastic modeling, decision fusion, deep learning, ensemble learning, and the EM algorithm. The overall diagrams of the proposed approaches are shown in Fig. 1.

In this framework, neural networks and decision tree ensembles act as non-parametric estimators of posterior probability, thus allowing multimodal data fusion. Specifically, fully convolutional networks (FCN) [21] are employed to estimate the posterior probabilities on the image with the finest spatial resolution (i.e., the UAV acquisition), and random forest (RF) [4] on the image with the coarsest spatial resolution (i.e., the resolution of the satellite acquisition). Indeed, the ratio between the two spatial resolutions is very high. Therefore, even though the pixel lattice of the UAV image can be quite large, the corresponding satellite image is expected to be composed of relatively few pixels, hence generally unfit for deep learning methods. That is the rationale of the use of a decision tree ensemble to predict pixelwise posteriors on the pixel grid of the satellite image.

In general, the proposed approaches can be combined with an arbitrary FCN model. In particular, U-Net [34] is used as the reference model on the UAV lattice, since it is widely employed and has been found to be effective in applications to remote sensing imagery. Likewise, for the ensemble learning technique, RF was selected for its well-known computational efficiency and flexibility to model heterogeneous data.

After the computation of the pixelwise posterior probabilities of the multiresolution image pixel lattices by the FCN and RF, the first proposed method (see Sect. 3.1 and Fig. 1(a)) performs a pixelwise probabilistic fusion to obtain the final classification results exploiting the information carried by the UAV and the satellite imagery. A formulation of the EM algorithm allows to automatically estimate the transition probabilities that determine the chance of having a certain label at the finer spatial resolution given the label at the coarser spatial resolution.

For the second method (see Sect. 3.3 and Fig. 1(b)), on the other hand, the pixelwise posterior probabilities computed on the multiresolution image lattices

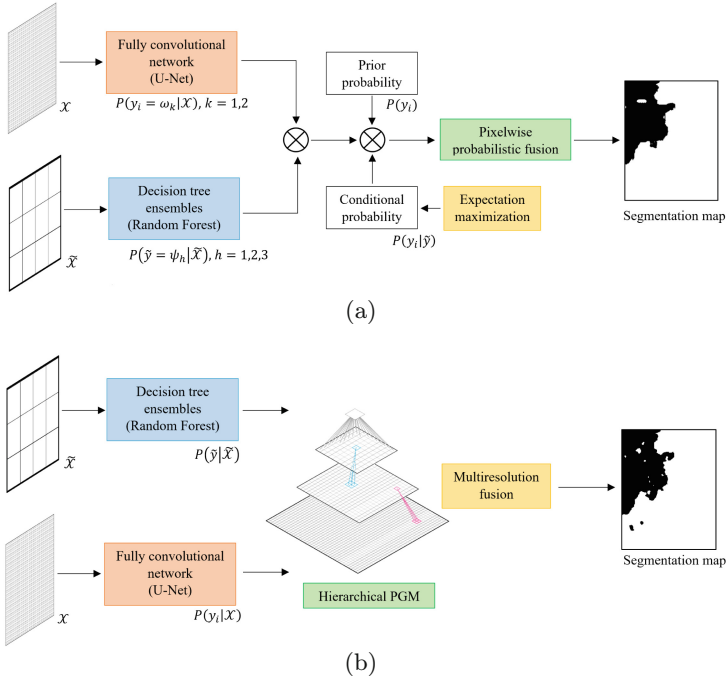


Fig. 1. Architecture of the two proposed methods: (a) pixelwise probabilistic fusion and (b) multiresolution fusion through hierarchical PGM.

at the native resolutions are fused through a hierarchical probabilistic graphical framework based on a hierarchical Markov random field, which models the multiresolution transition probabilities over a quadtree structure.

3.1 Pixelwise Probabilistic Fusion

The first proposed method introduces a pixelwise probabilistic decision fusion framework to combine the information contained at different resolutions [2, 5]. With the assumption that the two images are well registered, let us consider a patch of size $D \times D$ of the image at the finer resolution, with size corresponding exactly to one pixel in the lattice associated with the image at the coarser spatial resolution. Accordingly, D represents the resolution ratio associated with the input multiresolution dataset. The idea of the proposed method is to separately extract the thematic information contained in the two acquisitions collected at very different spatial resolutions and with generally different spectral bands, and perform a posterior probability pixelwise decision fusion.

As the ultimate task is to perform supervised binary image classification, the method requires a training map at both considered spatial resolutions. We assume that a training (ground truth, GT) map for two classes ω_1 and ω_2 is available for the acquisition at the finer spatial resolution. It is necessary to also

define classes and their training information on the coarser lattice. Focusing on the aforementioned $D \times D$ patch, this is determined through the following rules:

1. If all $D \times D$ finer-resolution pixels are training samples for ω_1 , then the corresponding coarser-resolution pixel is a training sample for class ψ_1 ;
2. If all $D \times D$ finer-resolution pixels are training samples for ω_2 , then the corresponding coarser-resolution pixel is a training sample for class ψ_2 ;
3. Else, the coarser resolution pixel is a training sample for class ψ_3 .

Accordingly, the two resolution levels correspond to distinct sets of classes: $\Omega = \{\omega_1, \omega_2\}$ on the finer resolution lattice and $\tilde{\Omega} = \{\psi_1, \psi_2, \psi_3\}$ on the coarser resolution grid. Semantically, ψ_1 and ω_1 represent the same land-cover class, but observed at the two very diverse resolutions – and the same comment holds about ψ_2 and ω_2 as well. On the contrary, ψ_3 represents a “mixed” class on the coarser-resolution lattice. The presence of this class is consistent with the fact that this pixel in the satellite image is necessarily a mixed pixel, corresponding to a ground area that is covered by partly ω_1 and partly ω_2 .

Let $x_i \in \mathbb{R}^n$ and $y_i \in \Omega$ be the feature vector and the class label, respectively, of the i th pixel of the $D \times D$ patch in the finer-resolution image, and let $\tilde{x} \in \mathbb{R}^m$ and $\tilde{y} \in \tilde{\Omega}$ be the feature vector and the class label, respectively, of the corresponding coarser-resolution pixel. We collect all finer-resolution feature vectors x_i within the patch in a tensor $X \in \mathbb{R}^{D \times D \times n}$. The first proposed method is formalized as follows in terms of a decision fusion approach from suitable input posteriors. Specifically, the posterior distribution of y_i , given all available input observations at both resolutions, i.e., given both X and \tilde{x} , can be expressed as:

$$P(y_i|X, \tilde{x}) = \sum_{\tilde{y} \in \tilde{\Omega}} P(y_i, \tilde{y}|X, \tilde{x}). \quad (1)$$

Applying the Bayes theorem:

$$\sum_{\tilde{y} \in \tilde{\Omega}} P(y_i, \tilde{y}|X, \tilde{x}) = \sum_{\tilde{y} \in \tilde{\Omega}} p(X, \tilde{x}|y_i, \tilde{y}) \frac{P(y_i, \tilde{y})}{p(X, \tilde{x})} \propto \sum_{\tilde{y} \in \tilde{\Omega}} p(X, \tilde{x}|y_i, \tilde{y}) P(y_i, \tilde{y}), \quad (2)$$

where $P(y_i, \tilde{y})$ is the pixelwise joint probability of the labels of the images at the two resolutions. The proportionality constant in (2) depends only on the features and not on the labels, hence it does not affect the decision. In the first proposed approach, we state the following conditional independence assumption:

$$p(X, \tilde{x}|y_i, \tilde{y}) = p(X|y_i)p(\tilde{x}|\tilde{y}). \quad (3)$$

Similar conditional independence assumptions are widely accepted in the development of Bayesian and Markovian approaches (e.g., in [12, 16, 18]). Under this assumption and considering again the Bayes theorem, plugging (3) into (2) implies:

$$P(y_i|X, \tilde{x}) \propto \sum_{\tilde{y} \in \tilde{\Omega}} P(y_i|X) \frac{p(X)}{P(y_i)} P(\tilde{y}|\tilde{x}) \frac{p(\tilde{x})}{P(\tilde{y})} P(y_i, \tilde{y}) \propto \sum_{\tilde{y} \in \tilde{\Omega}} P(y_i|X) P(\tilde{y}|\tilde{x}) \frac{P(y_i|\tilde{y})}{P(y_i)}, \quad (4)$$

where $P(y_i|X)$ is the posterior probability of the acquisition at finer spatial resolution conditioned on all feature vectors in the $D \times D$ patch, $P(\tilde{y}|\tilde{x})$ is the posterior probability computed for the acquisition at coarser spatial resolution on the individual pixel corresponding to the patch, $P(y_i)$ is the prior probability at the finer resolution, and $P(y_i|\tilde{y})$ is the conditional probability of the labels at the finer resolution given those at the coarser resolution.

Given the size of the input multiscale data, as mentioned in the previous section, $P(y_i|X)$ is estimated as the prediction $\hat{P}^{(\text{fcn})}(y_i|X)$ at the output of the softmax layer of the FCN and $P(\tilde{y}|\tilde{x})$ is predicted by the RF classifier in a pixelwise manner as $\hat{P}^{(\text{rf})}(\tilde{y}|\tilde{x})$.

Concerning the conditional probability $P(y_i|\tilde{y})$, first, stationarity is assumed. Specifically, for each pair (ω_k, ψ_h) of classes at the two resolutions, the joint probability $P\{y_i = \omega_k, \tilde{y} = \psi_h\}$ ($k = 1, 2; h = 1, 2, 3$) is assumed independent on the pixel location i . Therefore, the conditional probability $P\{y_i = \omega_k|\tilde{y} = \psi_h\}$ is independent of the location as well. Denoting $\theta_{k,h} = P\{y_i = \omega_k, \tilde{y} = \psi_h\}$, the joint probability matrix $\Theta = [\theta_{k,h}] \in \mathbb{R}^{2 \times 3}$ collects the parameters of the proposed method. Θ is estimated through an approximate formulation of the EM algorithm [7, 27], as explained in the next section.

3.2 EM-Based Estimation of the Transition Probabilities

EM is a well-known iterative method to address maximum-likelihood parameter estimation when the observations can be viewed as incomplete data [7, 27]. EM has been proved to converge (under mild assumptions) to a stationary point of the log-likelihood function [7, 41], although it does not converge, in general, to a global maximum point.

Let S be the coarser-resolution lattice. If $j \in S$ is a coarser-resolution pixel, \tilde{x}_j, \tilde{y}_j , and X_j are its feature vector, its label, and the corresponding finer-resolution tensor, respectively. The related $D \times D$ subset of the finer-resolution lattice is denoted as \mathcal{D}_j . Let \mathcal{X} be the tensor collecting all feature vectors x_i at the finer resolution and \mathcal{Y} be the matrix collecting all corresponding labels y_i ($\forall i \in \mathcal{D}_j, \forall j \in S$). Similarly, let $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ be the tensor of all feature vectors \tilde{x}_j and the matrix of all label \tilde{y}_j at the coarser resolution, respectively ($\forall j \in S$). EM iteratively maximizes the following function with respect to the matrix Θ of the parameters [7, 41]:

$$Q(\Theta|\Theta^t) = \mathbb{E} \left\{ \ln p(\mathcal{X}, \tilde{\mathcal{X}}, \mathcal{Y}, \tilde{\mathcal{Y}}|\Theta) \middle| \mathcal{X}, \tilde{\mathcal{X}}, \Theta^t \right\}, \quad (5)$$

where the superscript t is the iteration index ($t = 0, 1, 2, \dots$) and $p(\mathcal{X}, \tilde{\mathcal{X}}, \mathcal{Y}, \tilde{\mathcal{Y}}|\Theta)$ is the joint distribution of all feature vectors and labels, in which the dependence on Θ is explicitly emphasized. Equivalently:

$$Q(\Theta|\Theta^t) = \mathbb{E} \left\{ \ln p(\mathcal{X}, \tilde{\mathcal{X}}|\mathcal{Y}, \tilde{\mathcal{Y}}) + \ln P(\mathcal{Y}, \tilde{\mathcal{Y}}|\Theta) \middle| \mathcal{X}, \tilde{\mathcal{X}}, \Theta^t \right\}. \quad (6)$$

Here, Θ determines the joint distribution $P(\mathcal{Y}, \tilde{\mathcal{Y}}|\Theta)$ of all labels, whereas the probability density function $p(\mathcal{X}, \tilde{\mathcal{X}}|\mathcal{Y}, \tilde{\mathcal{Y}})$ of all observations, given all labels, is not parameterized on Θ and does not depend on it.

Specifically, in the proposed method, the function $Q(\Theta|\Theta^t)$ is replaced by an approximate formulation, in which we accept the following conditions:

1. The joint label distribution can be factored out as:

$$P(\mathcal{Y}, \tilde{\mathcal{Y}}|\Theta) = \prod_{j \in S} \prod_{i \in \mathcal{D}_j} P(y_i, \tilde{y}_j|\Theta); \quad (7)$$

2. For each coarser-resolution pixel $j \in S$, the label \tilde{y}_j and all labels y_i of the related finer-resolution pixels $i \in \mathcal{D}_j$ are independent on the observations associated with all other coarser-resolution samples $\tilde{x}_s, s \neq j$ and all the related finer-resolution samples $x_r, r \in \mathcal{D}_s$.

We recall that approximate formulations, based for instance on pseudo-likelihood or mean-field concepts, have been widely used in the application of EM-type algorithms to favor analytical feasibility or computational efficiency [14, 28, 44]. Here, conditions 1 and 2 are used precisely for this purpose, in the estimation of the parameters Θ . However, it is worth noting that such approximation is not involved at all in the training or prediction of the FCN and the RF classifiers.

Plugging (7) into (6), dropping the terms of (6) that do not depend on Θ , and applying condition 2 lead to the following approximate formulation:

$$\begin{aligned} \bar{Q}(\Theta|\Theta^t) &= \sum_{j \in S} \sum_{i \in \mathcal{D}_j} \mathbb{E} \left\{ \ln P(y_i, \tilde{y}_j|\Theta) \mid \mathcal{X}, \tilde{\mathcal{X}}, \Theta^t \right\} \\ &= \sum_{j \in S} \sum_{i \in \mathcal{D}_j} \mathbb{E} \left\{ \ln P(y_i, \tilde{y}_j|\Theta) \mid X_j, \tilde{x}_j, \Theta^t \right\}. \end{aligned} \quad (8)$$

Since $\theta_{k,h} = P\{y_i = \omega_k, \tilde{y}_j = \psi_h\}$ ($i \in \mathcal{D}_j$), we can write explicitly:

$$\bar{Q}(\Theta|\Theta^t) = \sum_{j \in S} \sum_{i \in \mathcal{D}_j} \sum_{k=1}^2 \sum_{h=1}^3 \alpha_{i,k,h}^t \ln \theta_{k,h}, \quad (9)$$

where $\alpha_{i,k,h}^t = P\{y_i = \omega_k, \tilde{y}_j = \psi_h | X_j, \tilde{x}_j, \Theta^t\}$ ($i \in \mathcal{D}_j; j \in S; k = 1, 2; h = 1, 2, 3; t = 0, 1, 2, \dots$) and where:

$$\sum_{k=1}^2 \sum_{h=1}^3 \theta_{k,h} = 1. \quad (10)$$

The updated parameter matrix Θ^{t+1} is obtained by maximizing the function $\bar{Q}(\Theta|\Theta^t)$ in (9) with respect to Θ under the constraint in (10). Solving the maximization through the Lagrangian multipliers, we obtain, after straightforward algebraic calculations ($k = 1, 2; h = 1, 2, 3; t = 0, 1, 2, \dots$):

$$\theta_{k,h}^{t+1} = \frac{1}{|S|D^2} \sum_{j \in S} \sum_{i \in \mathcal{D}_j} \alpha_{i,k,h}^t, \quad (11)$$

where $|S|$ is the total number of pixels in the coarser-resolution lattice (i.e., the cardinality of S). Owing to the conditional-independence assumption in (3), one can also prove that ($i \in \mathcal{D}_j; j \in S; t = 0, 1, 2, \dots$):

$$\alpha_{i,k,h}^t = A_i^t \frac{\theta_{k,h}^t P\{y_i = \omega_k | X_j, \Theta^t\} P\{\tilde{y}_j = \psi_h | \tilde{x}_j, \Theta^t\}}{\left(\sum_{\ell=1}^2 \theta_{\ell,h}^t\right) \left(\sum_{\ell=1}^3 \theta_{k,\ell}^t\right)}, \quad (12)$$

where A_i^t is a normalization constant that ensures that $\sum_{k=1}^2 \sum_{h=1}^3 \alpha_{i,k,h}^t = 1$. In the proposed method, we evaluate $\alpha_{i,k,h}^t$ by estimating the posteriors in the numerator of (12) as in the previous section, i.e., through their predictions $\hat{P}^{(\text{fcn})}(y_i | X_s)$ and $\hat{P}^{(\text{rf})}(\tilde{y}_s | \tilde{x}_s)$ computed by the FCN on the finer-resolution lattice and by RF on the coarser-resolution one, respectively:

$$\alpha_{i,k,h}^t = A_i^t \frac{\theta_{k,h}^t \hat{P}^{(\text{fcn})}\{y_i = \omega_k | X_j\} \hat{P}^{(\text{rf})}\{\tilde{y}_j = \psi_h | \tilde{x}_j\}}{\left(\sum_{\ell=1}^2 \theta_{\ell,h}^t\right) \left(\sum_{\ell=1}^3 \theta_{k,\ell}^t\right)}, \quad (13)$$

The approximate EM algorithm, integrated in the proposed method for the estimation of the joint pixelwise probabilities Θ of the labels at the two resolutions, is initialized with a uniform distribution Θ^0 (i.e., $\theta_{k,h}^0 = 1/6$ for $k = 1, 2$ and $h = 1, 2, 3$). Then, it iteratively alternates (13) and (11) until convergence.

3.3 Multiresolution Fusion Through Hierarchical Probabilistic Graphical Model

The second proposed method aims to perform the fusion of the multimodal multiresolution information through a pyramidal tree structure, where the imagery can be inserted, modeled, and analyzed at its native resolution (see Fig. 2).

In this case, the root level (level 0) of the tree contains the coarse-resolution image and the leaf level (level L) contains the fine-resolution image. Accordingly, each root pixel corresponds to $D \times D$ leaf pixels. Starting from the leaf level, intermediate levels $(L-1), \dots, 2, 1$ are constructed as in a traditional quadtree, by progressively halving the spatial resolution, and by associating the intermediate activations of the neural network at the corresponding resolution. Then, the root, i.e., level 0 of the tree, is linked directly to level 1. Differently from a conventional quadtree, where each consecutive level has a power-of-two relationship with the previous one, here, many more connections are present between the root and level 1. In particular, each pixel at the root corresponds to a patch of $(2^{1-L}D) \times (2^{1-L}D)$ pixels on level 1.

A hierarchical probabilistic graphical model (PGM) is defined over this pyramidal tree. As compared to PGMs on hierarchical quadtrees, this partially irregular topology affects the formulation of the inference criterion and the top-down and bottom-up flow of information across the levels.

Specifically, let S^ℓ be the pixel lattice of level ℓ of the tree ($\ell = 0, 1, \dots, L$)¹. We focus again, like in Sect. 3.1, on a single individual coarse-resolution pixel

¹ In Sect. 3.2, the pixel lattice of the input coarser-resolution image was indicated S . Here, it is denoted S^0 to distinguish it from the other lattices in the tree.

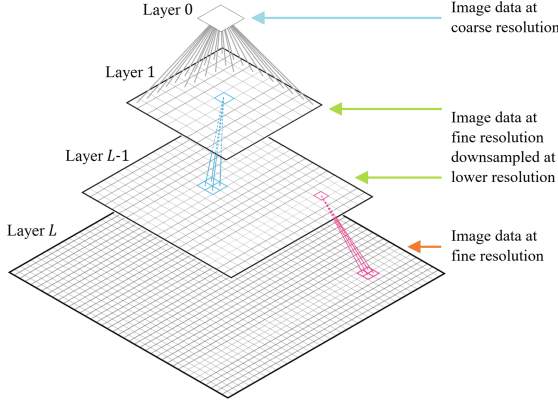


Fig. 2. Architecture of the MPM information fusion based on a quadtree.

(i.e., a single root pixel) and on the corresponding $D \times D$ patch at the fine resolution (i.e., at the leaves), and we use the same notations $\tilde{x}, \tilde{y}, x_i, y_i$, and X for the observations and labels at the root and at the leaves ($i \in S^L$). Here, we extend the notation x_i and y_i to the feature vector and the label of a pixel $i \in S^\ell$ also in an intermediate level of the tree ($\ell = 1, 2, \dots, L-1$).

The inference is performed through the marginal posterior mode (MPM) criterion [12]. Similar to [30], under suitable conditional independence assumptions MPM can be formulated on the proposed pyramidal tree as follows (the proof is omitted for brevity). Firstly, a top-down pass to compute the prior probability of the class label, starting from the root to the leaves is performed. From the root to level 1, this implies:

$$P(y_i) = \sum_{\tilde{y} \in \tilde{\Omega}} P(y_i|\tilde{y})P(\tilde{y}) \quad \forall i \in S^1. \quad (14)$$

Then, from level 1 to the leaves:

$$P(y_i) = \sum_{y_i^- \in \Omega} P(y_i|y_i^-)P(y_i^-) \quad \forall i \in S^\ell, \ell = 2, 3, \dots, L, \quad (15)$$

where $i^- \in S^{\ell-1}$ denotes the parent of a pixel $i \in S^\ell$ not on the root ($\ell > 0$).

Secondly, a bottom-up pass is performed from the leaves to the root to compute the distribution of the label y_i of each pixel i , given all observations of the descendants of i in the tree (collected in a vector x_i^d) [30]:

$$P(y_i|x_i^d) \propto P(y_i|x_i) \prod_{r \in i^+} \sum_{y_r \in \Omega} \frac{P(y_r|x_r^d)P(y_r|y_i)}{P(y_r)}, \quad (16)$$

$$P(y_i|y_i^c, x_i^d) \propto \frac{P(y_i|x_i^d)P(y_i|y_i^-)P(y_i^-)}{P(y_i)^{n_i}} \quad \forall i \in S^\ell, \ell = L-1, L-2, \dots, 0$$

where $i^+ \subset S^{\ell+1}$ is the set of the children of a pixel $i \in S^\ell$ not on the leaves ($\ell < L$), y_i^c collects the labels of all pixels connected to i in the tree, and n_i is the number of such pixels. Finally, a second top-down pass is performed to compute $P(y_i|X, \tilde{x})$ on all pixels i in the tree [30]:

$$P(y_i|X, \tilde{x}) = \sum_{y_i^c \in \Omega^{n_i}} P(y_i^c|y_i, x_i^d) P(\tilde{y}|X, \tilde{x}) \quad \forall i \in S^1, \quad (17)$$

$$P(y_i|X, \tilde{x}) = \sum_{y_i^c \in \Omega^{n_i}} P(y_i^c|y_i, x_i^d) P(y_{i-}|X, \tilde{x}) \quad \forall i \in S^\ell, \ell = 2, 3, \dots, L. \quad (18)$$

Accordingly, a pixel $i \in S^L$ is assigned the label that maximizes $P(y_i|X, \tilde{x})$.

More details can be found in [30]. On the leaf level the predictions $\hat{P}^{(\text{fcn})}(y_i|X)$ of the FCN on the finer-resolution image are used and incorporated in the PGM through (16). On the root, the predictions $\hat{P}^{(\text{rf})}(\tilde{y}|\tilde{x})$ from RF on the coarser-resolution image are used. On the intermediate levels, the pixelwise posteriors are computed through a softmax over the intermediate activations of the FCN, after a pass through a convolutional layer whose number of filters is equal to $|\Omega|$. Accordingly, in the proposed approach, the hierarchical PGM on the pyramidal tree addresses multiresolution fusion, merging the predictions from the deep neural and ensemble components.

4 Experimental Results

The proposed methods were tested on a multiresolution dataset for burnt area mapping in case of wildfires. The dataset consists of an RGB image acquired by an UAV with a spatial resolution of about 2 cm and the NIR channels of a Sentinel-2 image with a spatial resolution of 10 m (Fig. 3(a)-(b)). In particular, the UAV image has size of 16904×20324 pixels. Given the relationship between resolutions, $D = 480$. To maintain a reasonable number of levels and, simultaneously, model multiscale information, the drone imagery was resized to 4 cm and 8 cm of resolution (i.e., $L = 3$). Hence, each Sentinel-2 pixel is the parent of the 14400 pixels of the layer at 8 cm of resolution (with size 120×120 pixels).

The study area is La Destrousse, Provence-Alpes-Côte d’Azur, France. The drone image was acquired by INRAE (Institut National de Recherche pour l’Agriculture, l’Alimentation et l’Environnement) Provence-Alpes-Côte d’Azur research centre, Aix-en-Provence, shortly after the fire of 11 July 2018. The first available Sentinel-2 image of the same zone is dated 14 July 2018.

The GT boundaries of the burnt area, provided by the experts, were found with the canopy height model (CHM), measuring the height of trees, buildings, and other structures above the ground topography [11] (see Fig. 3(c)). This dataset was properly split in separate zones for training and testing the two proposed methods (see Fig. 4(a)).

To our knowledge, the proposed approaches are the first ones combining multiresolution UAV and satellite images at their - very different - native resolutions,

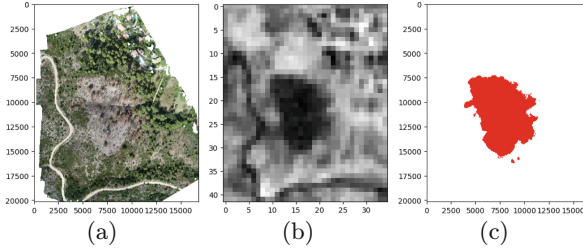


Fig. 3. Input images and GT: (a) drone image at 2 cm resolution, (b) Sentinel-2 image at 10 m resolution (the normalized difference vegetation index, NDVI, is displayed), (c) the GT with the same resolution as of the drone image.

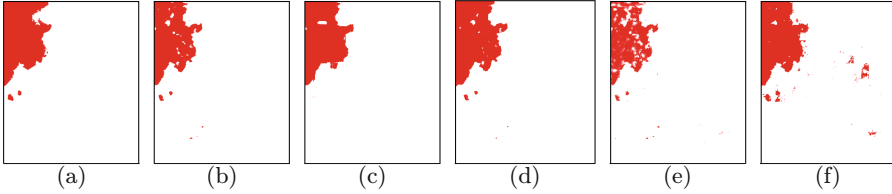


Fig. 4. GT and classification results on the UAV image: (a) GT used for testing (crop of Fig. 3(c)), and the classification results from (b) U-Net trained on UAV data, (c) the first and (d) the second proposed methods, (e) deep learning multiresolution fusion used for comparison, and (f) DBINet [9] trained on UAV data. Class legend: burnt (red) and non-burnt (white). (Color figure online)

for the mapping of burnt areas, therefore comparisons with state-of-the-art methods developed for this specific task were not possible. Nevertheless, the results of the proposed approaches were compared with those of the baseline U-Net, trained on the drone image at fine resolution, with those of RF trained on the satellite data at coarse resolution, and with a deep learning multiresolution fusion architecture where the satellite data at coarse resolution are included in the first convolutional layer as a bias scalar term given by the spectral information of the pixel \tilde{x} multiplied by a learnable weighting vector.

The method was also compared with a recent state-of-the-art approach for burnt area segmentation combining CNNs and transformers, DBINet [9]. Since the methodology does not involve a multiresolution input, it was trained either with the UAV very-high-resolution data or with the satellite imagery.

The quantitative results obtained by the proposed methods and the two approaches used for comparison are reported in Table 1 in terms of false and missed alarm rates, and overall error rate with respect to the GT test tile. The classification maps are shown in Fig. 4(b)-(f). On the one hand, the baseline U-Net directly applied to the UAV image is quite effective in the discrimination of burnt and non-burnt areas, as suggested by the results shown in Fig. 4(b) with an overall error rate equal to 1.61%. The same can be said for DBINet [9], which

Table 1. Classification accuracies of the proposed methods and of the comparison techniques.

Architecture	False alarm rate	Missed alarm rate	Overall error rate
U-Net on UAV [34]	0.28	10.81	1.61
RF on Sentinel-2 [4]	0.27	20.34	3.17
DL multires. fusion	0.19	27.08	3.51
First proposed method	0.48	8.55	1.47
Second proposed method	0.19	7.43	1.17
DBINet on UAV [9]	1.01	6.29	1.66
DBINet on Sentinel-2 [9]	0.69	12.45	2.15

attains similar results in terms of overall error rate, 1.66%. However, despite the output classification maps following the silhouette of the original GT map, there are several missed alarms inside the burnt zone for U-Net. On the other hand, the proposed pixelwise probabilistic fusion combining a few centimeters very high resolution RGB drone image and a multispectral satellite image with ten-meter resolution shows more accurate results in terms of overall error rate, alas with a small loss in terms of false alarm rate with respect to U-Net, and a small loss in terms of missed alarm rate with respect to DBINet. On the contrary, DBINet presents the highest false alarm rate, 1.01%, thus several false positives, consistently with low missed alarm rate (see Fig. 4(f)). The classification map of the first proposed method (Fig. 4(c)) is more visually smooth and accurate than the result of U-Net and DBINet, thanks to the integration of the multispectral Sentinel-2 data through the proposed approach.

The second proposed method, the multiresolution fusion through the hierarchical PGM, outputs the classification map after considering all the information of the observation of the descendant pixels and the labels of all the connected pixels. Thanks to this multiresolution multispectral information fusion, the method attains the best performances for the experimental validation with the UAV and satellite images processed at their native resolutions, in terms of all the accuracy metrics considered, except missed alarm rate. The overall error rate is slightly higher than 1% and the false alarm rate is about 0.2%. As compared to DBINet, it attains a slightly higher missed alarm rate, yet maintaining low values for both false positives and false negatives. The classification map shown in Fig. 4(d) confirms the potential of this proposed model, as it is visually smooth and accurate, especially in comparison with the original GT, outperforming not only the baseline but also the previous fusion method.

The results of the deep learning multiresolution fusion (see Table 1 and Fig. 4(e)) suggest its potential in mapping burnt areas, reaching the lowest false alarm rate of 0.19%, same as the second proposed method. However, the classification map and the performances in terms of missed alarm rate and overall error rate are poorer than those obtained by the two proposed techniques and U-Net.

The performances of the two methodologies trained on the satellite imagery, RF and DBINet (on Sentinel-2) appear to be suboptimal with respect to the ones obtained by the methodologies trained on the UAV imagery (U-Net and DBINet on UAV), or on the fusion of the two multiresolution inputs (the two proposed methods and the deep learning multiresolution fusion). This can be explained by the lower number of training samples and the coarser spatial resolution of the satellite imagery.

In general, the results in terms of missed alarm rate are worse than those of false alarm rate, due to the imbalance of the classes in the dataset, where “non-burnt” is clearly a majority class, thus prompting this behavior.

5 Conclusion

This paper introduced two probabilistic fusion methods for the joint use of multiresolution imagery with a big mismatch in spatial resolution in the framework of semantic segmentation tasks. In particular, the focus was on RGB images collected by UAV and multispectral satellite data, thus bringing to a resolution ratio between the input image sources of the order of the hundreds.

The methods were applied to a case study of wildfire burnt zones semantic segmentation and experimentally validated with a real dataset consisting of drone and Sentinel-2 image data collected over the South of France. The experiments show the effectiveness of the two proposed methods for the detection and mapping of zones affected by fires. The two developed techniques obtain accurate classification results and maps, in particular for the approach fusing multiresolution information through an irregular quadtree topology, a hierarchical PGM, and an FCN. This confirms the potential of the combination of FCN architectures with PGMs on appropriate graphs.

Perspectives for future developments will involve the integration of the proposed methodologies with transfer learning techniques to test it with image data acquired by different sensors, thus characterized by different features, and associated with different geographical areas. Furthermore, it would be interesting to apply the method to different case studies related to other applications involving multiresolution input imagery with great mismatch in spatial, and possibly spectral, resolution.

References

1. Alvarez-Vanhard, E., Corpetti, T., Houet, T.: UAV & satellite synergies for optical remote sensing applications: a literature review. *Sci. Remote Sens.* **3**, 100019 (2021)
2. Benediktsson, J., Kanellopoulos, I.: Classification of multisource and hyperspectral data based on decision fusion. *IEEE Trans. Geosci. Remote Sens.* **37**(3), 1367–1377 (1999)
3. Bouman, C., Liu, B.: Multiple resolution segmentation of textured images. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(2), 99–113 (1991)
4. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)

5. Briem, G., Benediktsson, J., Sveinsson, J.: Multiple classifiers applied to multi-source remote sensing data. *IEEE Trans. Geosci. Remote Sens.* **40**(10), 2291–2299 (2002)
6. Cole-Rhodes, A., Johnson, K., LeMoigne, J., Zavorin, I.: Multiresolution registration of remote sensing imagery by optimization of mutual information using a stochastic gradient. *IEEE Trans. Image Process.* **12**, 1495–1511 (2003)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Series B (Stat. Methodol.)* **39**(1), 1–38 (1977)
8. Esch, T., et al.: Breaking new ground in mapping human settlements from space - the global urban footprint. *ISPRS J. Photogramm. Remote. Sens.* **134**, 30–42 (2017)
9. Fang, W., Fu, Y., Sheng, V.S.: Dual backbone interaction network for burned area segmentation in optical remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **21**, 1–5 (2024)
10. Gómez-Chova, L., Tuia, D., Moser, G., Camps-Valls, G.: Multimodal classification of remote sensing images: a review and future directions. *Proc. IEEE* **103**(9), 1560–1584 (2015)
11. Hyyppä, J., Hyyppä, H., Leckie, D., Gougeon, F., Yu, X., Maltamo, M.: Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests. *Int. J. Remote Sens.* **29**(5), 1339–1366 (2008)
12. Kato, Z., Zerubia, J.: Markov random fields in image segmentation. *Found. Trends Signal Process.* **5**(1–2), 1–155 (2012)
13. Kim, J.B., Kim, H.J.: Multiresolution-based watersheds for efficient image segmentation. *Pattern Recognit. Lett.* **24**(1), 473–488 (2003)
14. Kuhn, E., Matias, C., Rebafka, T.: Properties of the stochastic approximation EM algorithm with mini-batch sampling. *Stat. Comput.* **30**(6), 1725–1739 (2020). <https://doi.org/10.1007/s11222-020-09968-0>
15. Laferté, J.M., Heitz, F., Perez, P.: A multiresolution EM algorithm for unsupervised image classification. In: *International Conference on Pattern Recognition (ICPR)*. vol. 2, pp. 849–853 (1996)
16. Laferté, J.M., Pérez, P., Heitz, F.: Discrete Markov image modeling and inference on the quadtree. *IEEE Trans. Image Process.* **9**(3), 390–404 (2000)
17. Laine, A., Fan, J.: Texture classification by wavelet packet signatures. *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(11), 1186–1191 (1993)
18. Li, S.Z.: *Markov random field modeling in image analysis*. Springer, 3rd edn. (2009)
19. Liu, H., et al.: Clusterformer for pine tree disease identification based on UAV remote sensing image segmentation. *IEEE Trans. Geosci. Remote Sens.* **62**, 1–15 (2024)
20. Liu, S., Cai, T., Tang, X., Wang, C.: MRL-Net: multi-scale representation learning network for COVID-19 lung CT image segmentation. *IEEE J. Biomed. Health Inform.* **27**(9), 4317–4328 (2023)
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440 (2015)
22. Luan, W., Zhang, X., Xiao, P., Wang, H., Chen, S.: Binary and fractional MODIS snow cover mapping boosted by machine learning and big Landsat data. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–14 (2022)
23. Luotamo, M., Metsämäki, S., Klami, A.: Multiscale cloud detection in remote sensing images using a dual convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **59**(6), 4972–4983 (2021)

24. Luppino, L.T., et al.: Code-aligned autoencoders for unsupervised change detection in multimodal remote sensing images. *IEEE Trans. Neural Netw. Learn. Syst.* **35**(1), 60–72 (2024)
25. Mallat, S.: A wavelet tour of signal processing – The sparse way. Academic Press, 3rd edn. (2009)
26. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(7), 3523–3542 (2022)
27. Moon, T.: The expectation-maximization algorithm. *IEEE Signal Process. Mag.* **13**(6), 47–60 (1996)
28. Moser, G., Serpico, S.B.: Unsupervised change detection from multichannel SAR data by Markovian data fusion. *IEEE Trans. Geosci. Remote Sens.* **47**(7), 2114–2128 (2009)
29. Pastorino, M., et al.: Multisensor and multiresolution remote sensing image classification through a causal hierarchical Markov framework and decision tree ensembles. *Remote Sens.* **13**(5), 849 (2021)
30. Pastorino, M., Moser, G., Serpico, S.B., Zerubia, J.: Semantic segmentation of remote-sensing images through fully convolutional neural networks and hierarchical probabilistic graphical models. *IEEE Trans. Geosci. Remote Sens.* **60**(5407116), 1–16 (2022)
31. Pyun, K., Lim, J., Won, C.S., Gray, R.M.: Image segmentation using hidden Markov Gauss mixture models. *IEEE Trans. Image Process.* **16**(7), 1902–1911 (2007)
32. Rezaee, M., van der Zwet, P., Lelieveldt, B., van der Geest, R., Reiber, J.: A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering. *IEEE Trans. Image Process.* **9**(7), 1238–1248 (2000)
33. Richards, J.A.: Remote sensing digital image analysis: An introduction. Springer, 5th edn. (2013)
34. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) *MICCAI 2015*. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
35. Song, P., Li, J., An, Z., Fan, H., Fan, L.: CTMFNet: CNN and transformer multi-scale fusion network of remote sensing urban scene imagery. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–14 (2023)
36. Thoonen, G., Mahmood, Z., Peeters, S., Scheunders, P.: Multisource classification of color and hyperspectral images using color attribute profiles and composite decision fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **5**(2), 510–521 (2012)
37. van Rijthoven, M., Balkenhol, M., Silina, K., van der Laak, J., Ciompi, F.: HookNet: multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images. *Med. Image Anal.* **68**, 101890 (2021)
38. Wang, L., Liu, J.: Texture classification using multiresolution Markov random field models. *Pattern Recognit. Lett.* **20**(2), 171–182 (1999)
39. Wang, L., Zhang, C., Li, R., Duan, C., Meng, X., Atkinson, P.M.: Scale-aware neural network for semantic segmentation of multi-resolution remote sensing images. *Remote Sens.* **13**(24), 5015 (2021)
40. Waske, B., Benediktsson, J.A.: Fusion of support vector machines for classification of multisensor data. *IEEE Trans. Geosci. Remote Sens.* **45**(12), 3858–3866 (2007)
41. Wu, C.F.J.: On the convergence properties of the EM algorithm. *Annal. Stat.* **11**(1), 95–103 (1983)

42. Zavorin, I., Moigne, J.: Use of multiresolution wavelet feature pyramids for automatic registration of multisensor imagery. *IEEE Trans. Image Process.* **14**, 770–82 (2005)
43. Zheng, G., Jiang, Z., Zhang, H., Yao, X.: Deep semantic segmentation of unmanned aerial vehicle remote sensing images based on fully convolutional neural network. *Front. Earth Sci.* **11**, 1115805 (2023)
44. Zhou, F., et al.: Efficient inference for dynamic flexible interactions of neural populations. *J. Mach. Learn. Res.* **23**(211), 1–49 (2022)



Multiscale Color Guided Attention Ensemble Classifier for Age-Related Macular Degeneration Using Concurrent Fundus and Optical Coherence Tomography Images

Pragya Gupta¹(✉), Subhamoy Mandal², Debashree Guha²,
and Debjani Chakraborty¹

¹ Department of Mathematics, Indian Institute of Technology Kharagpur,
West Bengal 721302, India

g.pragya1@gmail.com, debjani@maths.iitkgp.ac.in

² School of Medical Science and Technology, Indian Institute of Technology
Kharagpur, West Bengal 721302, India

{smandal,debashree_smst}@smst.iitkgp.ac.in

Abstract. Automatic diagnosis techniques have evolved to identify age-related macular degeneration (AMD) by employing single modality Fundus images or optical coherence tomography (OCT). To classify ocular diseases, Fundus and OCT images are the most crucial imaging modalities used in the clinical setting. Most deep learning-based techniques are established on a single imaging modality, which contemplates the ocular disorders to a specific extent and disregards other modality that comprises exhaustive information among distinct imaging modalities. This paper proposes a modality-specific multiscale color space embedding integrated with the attention mechanism based on transfer learning for classification (MCGAEc), which can efficiently extract the distinct modality information at various scales using the distinct color spaces. In this work, we first introduce the modality-specific multiscale color space encoder model, which includes diverse feature representations by integrating distinct characteristic color spaces on a multiscale into a unified framework. The extracted features from the prior encoder module are incorporated with the attention mechanism to extract the global features representation, which is integrated with the prior extracted features and transferred to the random forest classifier for the classification of AMD. To analyze the performance of the proposed MCGAEc method, a publicly available multi-modality dataset from Project Macula for AMD is utilized and compared with the existing models.

Keywords: Fundus image · Optical coherence tomography · Multi-modality images · Classification · Transfer learning

1 Introduction

Diagnosing retinal disorders plays a vital role in guiding treatment decisions and improving outcomes for individuals with retinal conditions. Age-related maculopathy is a degenerative condition of the central region of the retina that is correlated with the cause of visual impairment that is recurring after 65 years of age [3]. The diagnosis of AMD was first described in [30]. In the earlier phases of AMD, the patients have drusen and RPE abnormalities, whereas geographic atrophy and neovascularization of the retina may be interconnected with vision loss during the progression of the disorder. AMD can be characterized as dry or wet according to the pathogenesis. Choroidal neovascularization (CNV) [14] is a manifestation of wet AMD and is diagnosed by analyzing the uncharacteristic expansion of blood vessels from the choroid into the retina. DME [16] is a serious condition that can be attributed to hyperglycemia, which is a form of diabetic retinopathy (DR). It occurs due to prolonged exposure to high blood sugar, particularly in diabetic patients, which causes fluid leakage into the macula region, swelling, and thickening. Drusen [2] is a condition of dry AMD where tiny yellow or white deposits accumulate under the retina. If not characterized immediately, these ocular disorders can impair the retinal layer, especially the macular area, and perhaps end up with vision loss. Traditional diagnostic and grading systems for AMD are conducted by analyzing the color Fundus images [12]. Over the years, significant advancements have been made in medical diagnostics, offering new tools and techniques that enable precise and early detection of retinal disorders, especially the evolution of imaging techniques using Fundus images. OCT is one of the widely exploited diagnostic tools that will provide 3D structural information associated with the demonstration of cross-sectional images. OCT provides constructive information in investigating retinal disorders in challenging diagnostic cases and acquiring cross-sectional lesions of neovascularization associated with neighborhood tissue information. Clinical practitioners utilize OCT to examine the activity of AMD nowadays [41]. These techniques are effective; however, they suffer from intrinsic constraints, including specialized clinical practitioners and time-consuming, which induce variation in the ocular diagnosis, and hindered intervention can emerge. Recent advancements in deep learning techniques have presented enormous possibilities for automated diagnosis tasks of retinal disorders at the expert level, decreasing the dependency on human experts in diverse fields [44]. There is a rapidly growing interest in implementing deep learning models for classification using imaging modalities such as color Fundus [29] or OCT images [9, 17]. Morano et al. [27] proposed an explainable weakly-supervised technique for AMD diagnosis by operating on color Fundus images. Philippe et al. [4] proposed to utilize Generative Adversarial Networks to obtain the synthetic image dataset for AMD classification. Tak et al. [38] proposed a classification model based on a convolutional neural network (CNN) to categorize exudative and non-exudative classes. Researchers have introduced diverse types of deep learning models by extracting the distinct kind of feature characterization for distinguishing the abnormalities emerging in the Fundus images for identifying the distinct stages of the AMD [1, 10]. However, the diag-

nosis of AMD by analyzing the Fundus images and ignoring additional kinds of investigation is restricted due to the 2D representation of the Fundus image. The Fundus image is independent of slight changes in the macular breakage, thickness, and detachment of the retinal layers [31]. On the other hand, OCT acquires a cross-sectional representation of biological tissues at microscopic spatial resolution [39], and it is a non-invasive technique. Liu et al. [24] proposed global feature image descriptors constructed based on machine learning for classifying four categories of AMD using OCT images. Karri et al. [17] introduced a classification technique based on transfer learning using OCT images. Sun et al. [37] employed automatically align and crop of retina area followed by global illustrations by utilizing sparse coding, and finally, a multiscale support vector machine is executed for AMD classification. Although various automatic techniques have been introduced for the diagnosis of AMD and their different classes for examining the severity level based on OCT or Fundus images. However, it is difficult to interpret intricate oculopathy with several lesions in the retina using a single imaging modality. Clinical practitioners generally consider two imaging modalities, as shown in Fig. 1, including color Fundus and OCT, in analyzing and diagnosing retinal disorders. Further, they simultaneously assess OCT and Fundus images and incorporate their specific feature representation details to provide accurate diagnoses. In this regard, several methods are introduced for the classification of ocular disorders using multi-modality imaging techniques [20]. Yoo et al. [43] employed a random forest classifier with a VGG model for the classification of AMD using multi-modality OCT and Fundus images. Yin et al. [7] proposed a TransMed method using multi-modality medical images for the classification tasks. Li et al. [23] used multi-modal evidence and introduced a multi-instance deep learning model for the diagnosis of retinal disorders. Wang et al. [40] proposed a two-stream CNN model for the classification of ocular diseases using Fundus and OCT images. Fang et al. [11] proposed a technique by integrating the regression approach to the deep learning models for the diagnosis of glaucoma grading. Xing et al. [42] proposed a transformer-based model using multi-modality images. These introduced deep learning models based on multi-modality Fundus and OCT images improve the performance of diagnosis of retinal disorders compared to the utilization of single modality imaging methods. Notably, relying on the single color channel and scale space of the multi-modality images employed for the deep learning models often ignores capturing the global and local feature representation in distinct multi-scale space. Moreover, the feature information in the distinct color spaces provides diverse characteristic feature information that is not attainable by employing a single color space.

In this work, we propose modality-specific multiscale color space embedding based on the attention mechanism for the classification of different stages of the AMD. The proposed framework includes the utilization of YCbCr and HSV color space for Fundus images on different scale spaces, followed by OCT images at multiscale to capture the crucial features by exploiting the different characteristic feature representations in distinct color spaces and fed to three

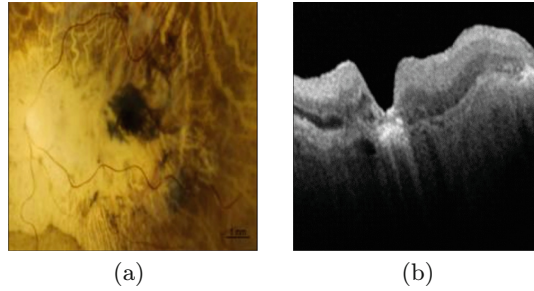


Fig. 1. (a) The Original Fundus image and (b) the corresponding OCT image.

different pre-trained VGG16 model for the feature extraction. It enhances the generalization ability of the proposed model and also considers the different feature attributes of the color spaces by considering distinct scales into a unified framework. An attention mechanism is incorporated to extract the most representative feature from each considered path, followed by a concatenation of the extracted features from each pre-trained model, and the attention mechanism is transferred to a random forest classifier (RFC) model for the classification of AMD. To experimentally analyze the interpretation of the proposed MCGAEc method, a publicly available multi-modality dataset is utilized, which is provided at Project Macula [32] for AMD and compared with the existing method to show the significance of the proposed framework. In summary, the key contributions to this study are mentioned below:

- We propose a modality-specific multiscale color space embedding integrated with the attention mechanism, which adaptively specifies the most representative features and is fed to the RFC model for the classification of AMD.
- To analyze the efficacy of the proposed MCGAEc model, experiments are performed over a publicly available dataset of the color Fundus and OCT images (Project Macula) and compared with the single modality (Fundus or OCT) images on different color spaces at distinct scales.
- The proposed method is compared with the state-of-the-art (SOTA) method to demonstrate the efficacy of the proposed MCGAEc model using the evaluation measures.

The remainder of the manuscript is systematized as follows: Sect. 2 demonstrates the proposed framework, followed by experimental analysis in Sect. 3. Section 4 includes the concluding observations.

2 Proposed Model

This section introduces the proposed MCGAEc framework illustrated in Fig. 2. MCGAEc comprises a modality-specific multiscale color space encoder module in which a Fundus image is transformed into distinct color spaces, say, YCbCr

and HSV, which are transferred to two encoder paths. For OCT images, one encoder path named gray-scale is constituted. In each path, the transformed Fundus images are forwarded to multiscale space conversion, followed by input for the pre-trained VGG16 model to acquire vital features from each considered color space at various scales. The extracted features from each path from the Fundus images are transferred to the self-attention module. Simultaneously, we extract the features from the gray-scale path for the OCT images. Then, features extracted from the pre-trained VGG16 and self-attention module are concatenated. Finally, we fused all the extracted features from each path and fed them to the RFC for the classification of AMD. **Modality-Specific Multiscale Color Space Encoder Model:** The color variations of Fundus images enclose exhaustive ranges, and heterogeneity in color casts restricts the classical models [6]. Motivated by traditional enhancement techniques that function over different color spaces [15, 25, 28], we extract distinct characteristic features from two color spaces (HSV, YCbCr) where the identical Fundus image has distinct pictorial representation in diverse color spaces demonstrated in Fig. 3. The Fundus image is explicitly to visualize in RGB color space because of its intense physical significance in color. However, the color segments R, G, and B are positively associated and are easy to be influenced by the variation of luminance, occlusion, and other factors. On the other hand, YCbCr color space can intuitively reminisce the luminance (Y) and two chroma components (Cb and Cr). YCbCr is crucial in digital images and video to separate luminance from its chrominance. This fragmentation is beneficial because the human eye is more susceptible to luminance than chrominance, and it entitles more efficiency in compression that aligns with the visual perception of humans. HSV color space characterizes the hue, saturation, contrast, and brightness of the Fundus image. The considered color space has diverse characteristics and benefits. To integrate their properties in the Fundus image feature enhancement, we assimilate the characteristics of distinct color spaces into a unified deep characteristic model. Furthermore, the color variations of two considered points with a diminutive variation in one color space can be enormous in another color space. Thus, the distinct color space integration can facilitate the measure of the color divergence of Fundus images.

To extract the necessary features from distinct color spaces, which is required for the classification of AMD, multi-scaling is incorporated. When the scale is augmented, most of the noise is eliminated. If the features are available in more than one coarse scale, which indicates it should be available at different scales as well [36]. Therefore, this strategy is implemented over each color space by considering different scales for Fundus and OCT images, respectively. Afterward, we executed the pre-trained VGG16 [35], which includes 16 deep layers for extracting the features from considered multiscale color space for Fundus and OCT images. VGG16 is an adequate model for the image classification task, and the pre-trained model enables to extraction of features from an extensive corpus of images when utilizing a small dataset. VGG16 model provides the balance between the performance and computational efficiency compared to other

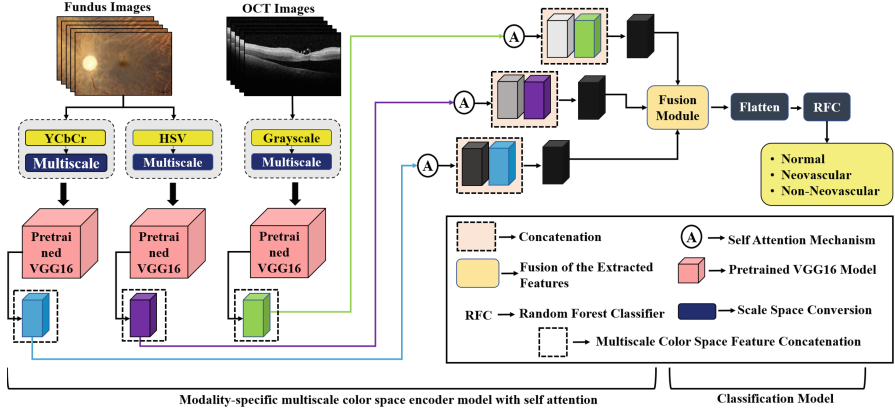


Fig. 2. The proposed framework for the combination of the Fundus and OCT images for the classification of AMD

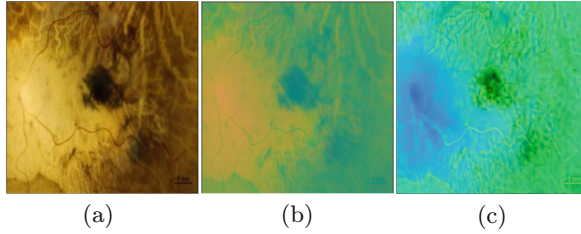


Fig. 3. Representation of the Fundus image in (a) RGB color space (original), (b) YCbCr color space, and (c) HSV color space.

complex models. Additionally, it allows the experiments without requiring comprehensive computational resources, which provides an advancement to integrate the attention layer with that. A transfer learning mechanism is utilized, and extracted features from each are fed to the distinct attention module.

Self-Attention: The self-attention mechanism was first introduced in the domain of image processing [33], which is integrated into the attention layer. It enables to concentrate on salient or global features of the datasets. It provides an adequate correlation towards global feature information within each single image. The primary notion behind the self-attention mechanism is to associate weighted average values evaluated from the prior layers, and the attention weights are assessed as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_{head}}}\right)V \quad (1)$$

where Q is a query, K is the key, and V is the value. In the assertion of the precise characterization of each multiscale color space feature, the extracted

features from the pre-trained model through each path have dissimilar contributions. Consequently, we utilize a self-attention mechanism to exploit the inter-variability between the extracted features obtained through each path. In the proposed framework, the extracted features from each path through the pre-trained VGG16 model are used as input for the self-attention module. The extracted features through the pre-trained VGG16 model are confined by structure to concentrate primarily on local features of Fundus and OCT images while incompetent to acquire the global feature information. Self-attention focuses on specific global features of the images. Therefore, the extracted features from the pre-trained deep network model are fed into the self-attention model corresponding to each path. The attention features and extracted prior features from the pre-trained model from each path are fused together by concatenation for a better representation of each feature acquired through the Fundus and OCT images, which are used as input for the classification model.

Classification Model: The computed features through the modality-specific multiscale color space encoder model with self-attention are transferred into the supervised machine learning RFC [19]. The RFC comprises an amalgamation of tree classifiers where a particular classifier is acquired through a random vector, which is sampled individually from the input data, and the respective tree provides a unit vote for the most prevalent class to categorize input data. The RFC is utilized for the classification of the input data, which relies on the utilization of randomly chosen features at each node to expand a tree. In RFC, features are illustrated on the inner nodes, which are called decision nodes, and enable to generate the predictions from a sequence of feature-based fragmentation. RFC employs a collection of random decision trees and integrates them concurrently to construct a robust model that is less acute to the training data. The RFC algorithm is capable of handling higher dimensional data and utilizing an enormous number of trees in the combination. The output of the RFC is computed by a majority counting of votes obtained through trees. Here, RFC is used, which includes considerably lower computational complexity as each particular tree exclusively utilizes a part of the input vector in a Random Forest.

3 Experimental Framework

In this section, the description of the multi-modality OCT and Fundus image datasets for diagnosis of AMD is provided, followed by evaluation measures to check the performance of the proposed MCGAEc model. Finally, a comprehensive empirical study, including the ablation analysis, is given to show the significance of the proposed MCGAEc model for the classification using the multi-modality dataset.

3.1 Dataset and Evaluation Measures

The experimental analysis of the MCGAEc model is assessed over the publicly open multi-modality OCT and Fundus images dataset at the Project

Macula [32] for AMD classification (<https://projectmacula.cs.uab.edu>). The publicly available dataset aimed to investigate AMD in patients and their severity level. The diagnosis of AMD over the provided dataset is endorsed by patho-histological examination. The dataset is categorized into three classes: normal, non-neovascular, and neovascular. The normal class includes 50 OCT and Fundus images each, followed by 19 and 40 in non-neovascular and neovascular, respectively.

The spatial resolution of Fundus and OCT images in each class is of different variation, and data is not enough to train the model. Therefore, data augmentation is applied with rotations, translations, and contrast changes for increments in the number of images. Data augmentation [34] is extensively applied to enhance the generalization of the proposed method. We erratically retrieved 500 Fundus images for each class and 500 OCT images for each class that matched with the Fundus images. We have performed the rotation in a range $[-25^\circ, +25^\circ]$, with translation $[-10\%, +10\%]$ of the width of the image, and contrast change with ranges of $[-50\%, +50\%]$. All the generated OCT and Fundus images are resized 224×224 for the input of the pre-trained model. The experiments are performed in the selection of an optimal number of trees from the set $\{100, 300, 500, 700, 1000\}$ and estimators from the range 3–25 for the RFC classifier. The optimal number of trees for the proposed framework for the RFC is to obtain 1000 trees and 10 estimators for each node for the AMD dataset. The five-fold cross-validation is employed for multi-classification, which is illustrated in Sect. 3.2

To examine the performance of the MCGAEc method on the above-considered dataset, commonly used performance evaluation measures, including AUC (area under the receiver operating characteristic), Accuracy, Sensitivity, Specificity, F_1 , and Matthews Correlation Coefficient (MCC) score is considered, which are described as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad Sensitivity = \frac{TP}{TP + FN} \quad (2)$$

$$Specificity = \frac{TN}{TN + FP} \quad F_1 = \frac{2TP}{2TP + FP + FN} \quad (3)$$

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4)$$

where TP denotes true positive counts, FN denotes false negative counts, TN denotes true negative counts, and FP denotes false positive counts. For all the evaluation measures, higher values indicate better classification performance.

3.2 Effectiveness of the Proposed MCGAEc Model

We first illustrate the comparison between the proposed MCGAEc model over the single modality by considering different color spaces with various scales. We have performed 5-fold cross-validation, and the experimental results are reported

Table 1. Experimental performance over the test set of the single modality model over the multi-modality model by considering different color spaces and scale space based on 5-fold cross-validation of the training set

Methods	AUC	Accuracy	Sensitivity	Specificity	F ₁	MCC
HSV+Fundus+RFC($\sigma = 1$)	0.970	0.870	0.870	0.935	0.869	0.810
YCbCr+Fundus+RFC($\sigma = 1$)	0.986	0.907	0.906	0.953	0.906	0.860
HSV+Fundus+multiscale+RFC	0.968	0.857	0.856	0.928	0.855	0.803
YCbCr+Fundus+multiscale+RFC	0.986	0.910	0.913	0.955	0.909	0.860
OCT+RFC($\sigma = 1$)	0.979	0.887	0.886	0.943	0.888	0.833
OCT+RFC($\sigma = 4$)	0.980	0.890	0.890	0.945	0.891	0.848
OCT+multiscale+RFC	0.980	0.893	0.894	0.947	0.894	0.847
YCbCr+HSV+Fundus+RFC($\sigma = 1$)	0.988	0.910	0.910	0.955	0.909	0.865
YCbCr+HSV+Fundus+RFC($\sigma = 4$)	0.987	0.907	0.907	0.953	0.906	0.879
YCbCr+Fundus+OCT+RFC($\sigma = 1$)	0.989	0.927	0.927	0.963	0.927	0.886
YCbCr+Fundus+OCT+RFC($\sigma = 4$)	0.990	0.930	0.932	0.965	0.930	0.895
HSV+Fundus+OCT+RFC($\sigma = 1$)	0.986	0.923	0.920	0.962	0.923	0.880
HSV+Fundus+OCT+RFC($\sigma = 4$)	0.987	0.916	0.917	0.958	0.916	0.861
HSV+YCbCr+Fundus+OCT+RFC($\sigma = 1$)	0.993	0.943	0.943	0.972	0.943	0.896
HSV+YCbCr+Fundus+OCT+RFC($\sigma = 4$)	0.992	0.937	0.932	0.968	0.937	0.892
Proposed (MCGAEc)	0.994	0.947	0.948	0.973	0.947	0.907

in Table 1. When considering a single modality Fundus image, we have performed the experiments on YCbCr and HSV color spaces by taking the lower and higher regularization levels to capture the features at different scales, as shown in Table 1. For OCT images, we have considered different scales for comparison with the proposed framework and other considered cases. The pre-trained VGG16 model is the backbone for the feature extraction from single modality and multi-modality with respect to different multi-scale color spaces.

Figure 4 demonstrates ROC curves for each considered case over the proposed framework for the AMD dataset. It can be observed from the ROC curve the proposed modality-specific multiscale color space embedding based on attention mechanism (MCGAEc model) for AMD classification is more adequate compared to other single modalities based on single color spaces. The proposed method achieves a higher AUC value of 0.994 compared to other single modalities with single color spaces, as demonstrated in Fig. 4. Table 1 signifies that the proposed MCGAEc model can improve the capability of AMD classification compared to the utilization of a single modality model. In the single modality, the Fundus image, when transformed to YCbCr color space with regularization level $\sigma = 4$, has achieved a higher AUC of 0.990 compared to other single modality cases. Further, it achieves a higher F₁-score of 0.930 among the other single modality for classification. However, when the Fundus image is transformed into HSV color space with multiscale, it achieves the lowest F₁ score compared with others. The proposed model has higher Accuracy, Sensitivity, Specificity, F₁, and MCC score compared to others demonstrated in Table 1. It indicates that

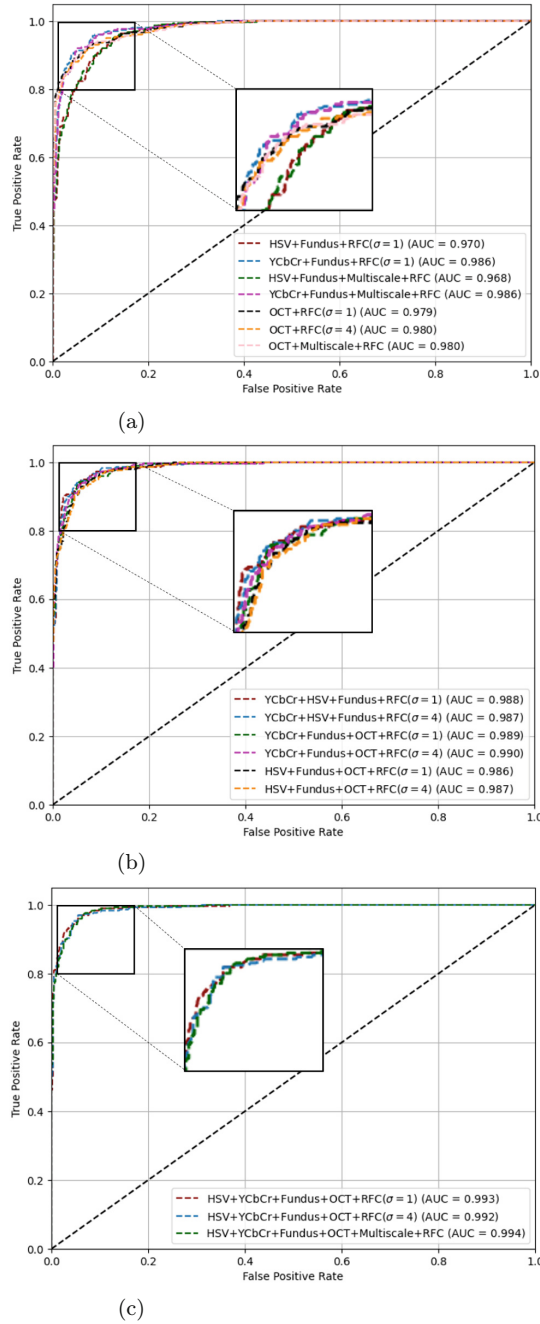


Fig. 4. ROC (Receiver operating characteristic) curves for the proposed method, (a) multiscale color space with single modality comparison, (b) fusion of the multiscale color spaces, (c) combination of multi-modality Fundus and OCT images for the multi-class classification of AMD.

Table 2. Experimental performance of the proposed MCGAEc model compared with the state-of-the-art method [43] over the training set using five-fold cross-validation for the AMD dataset

Methods	AUC	Accuracy	Sensitivity	Specificity	F ₁
RF - OCT image alone	0.906	0.826	0.833	0.816	—
RF - Fundus image alone	0.914	0.835	0.834	0.836	—
RF - OCT+Fundus image	0.969	0.905	0.910	0.896	—
RBM - OCT+Fundus image	0.940	0.865	0.860	0.875	—
DBN - OCT+Fundus image	0.956	0.889	0.880	0.905	—
Proposed (MCGAEc)	0.994	0.944	0.943	0.972	0.943

when different color spaces are considered in multiscale and incorporated with the multi-modality Fundus and OCT images, the performance of the classifier is enhanced significantly, and it enables to capture of essential local and global discriminative features at different scales using attention mechanism.

3.3 Quantitative Comparison with Existing Approaches

In this section, we have compared the proposed model with the existing method over the considered AMD dataset, and the empirical study is demonstrated in Table 2–3. For comparison purposes with the AMD dataset, the implementation of the SOTA methods is conducted. The training results over the AMD dataset of the proposed framework are presented in Table 2 and compared with the Yoo et al. [43], which indicates that the proposed model is competent to extract the most discriminative features, which is essential for the classification of AMD in comparison to [43]. The proposed MCGAEc model enables to distinguish between false positive and true positive at distinct threshold levels indicated by the higher AUC value acquired by the proposed framework. Table 3 represents the test result over the multi-modality Fundus and OCT image dataset and is compared with the SOTA methods. The proposed model attained a higher AUC of 0.994 compared to other methods, which indicates the consistency of the proposed classification framework at different threshold levels except [11], and it can be observed from the ROC curve illustrated in Fig. 4. The presented method [11] acquires lower Specificity compared to our proposed method, which indicates the diagnosis of AMD in the false positive category. MCGAEc model performance is adequate in terms of quantitative comparison for the considered AMD dataset. The empirical analysis of the proposed MCGAEc model is adequate compared to the SOTA approaches illustrated in Table 3 over the test set.

3.4 Discussion

This is the first experimental study, best to the knowledge of the authors to consider a multiscale color space for fusing distinct imaging modalities for the

classification of AMD disorder. In this study, we proposed an MCGAEc model that considers the Fundus and OCT images at multiscale color spaces simultaneously for the diagnosis of AMD. Fundus imaging modality characterizes information on the region of the drusen (AMD). OCT provides subsurface cross-section imaging, providing information about the different layers of the retina. Such information is complimentary to the structural information of the vasculature on the surface as obtained from Fundus images. OCT imaging modality is correlated with the subsurface of retinal layers and intra-retinal fluid lesions. The thickness of retinal layers, which is influenced by choroidal neovascularization, is investigated through the OCT, and the Fundus image is competent to apprehend the evolution in the size of the drusen. However, the Fundus image is insufficient in identifying choroidal neovascularization rigorously [26]. Whereas OCT is not able to identify the transitions in the drusen and retinal pigment epithelium [5]. The Fundus and OCT imaging modalities provide complementary information on the retina. Early stages-based techniques utilized Fundus or OCT for diagnosis of AMD based on deep learning models [8, 13, 18, 21, 22]. This work focuses on investigating the crucial feature information extracted from Fundus and OCT images, which are integrated with the proposed MCGAEc method to capture the surface and subsurface retinal information for the diagnosis of AMD and glaucoma. We have assimilated multiscale color space to diagnose complex retinal diseases, which helps to capture dissimilar features from OCT and Fundus simultaneously at distinct color spaces with multiple scales. In the proposed approach, we attempt to combine distinct feature properties preserved in the various color spaces and if it is captured at a finer scale so there is a possibility of the presence of that particular feature at multiple scales. Therefore, we have considered YCbCr and HSV color spaces at different scales and integrated them with the pre-trained VGG16 model to extract the crucial feature for diagnosis

Table 3. Experimental performance of the proposed MCGAEc model compared with the state-of-the-art method over the test set for the AMD dataset

Methods	AUC	Accuracy	Sensitivity	Specificity	F ₁
RF - OCT image alone [43]	0.914	0.833	0.808	0.883	—
RF - Fundus image alone [43]	0.954	0.892	0.900	0.877	—
RF - OCT+Fundus image [43]	0.981	0.946	0.955	0.927	—
RBM - OCT+Fundus image [43]	0.976	0.931	0.942	0.910	—
DBN - OCT+Fundus image [43]	0.961	0.892	0.888	0.913	—
Inception-v3 - Fundus image alone [43]	0.958	0.892	0.900	0.877	—
LASSO Regression - OCT+Fundus image [43]	0.950	0.885	0.887	0.882	—
ANN-Fundus segmentation [43]	0.911	0.841	0.852	0.820	—
Yin et al. [7]	0.998	0.938	0.937	0.969	0.936
Xin et al. [42]	0.991	0.925	0.925	0.963	0.924
Fang et al. [11]	0.996	0.945	0.941	0.951	0.946
Aiyub et al. [1]	0.989	0.945	0.939	0.952	0.946
Proposed (MCGAEc)	0.994	0.947	0.948	0.973	0.947

purposes. On the other hand, OCT images are considered on different scales to capture the subsurface retinal layer information for the diagnosis of choroidal neovascularization. The attention mechanism is incorporated to extract global feature representation and integrated with the local feature information, followed by the ensembling of each classifier at the feature fusion module, and a random forest classifier is utilized for the classification of various stages of AMD. To show the significance of the modality-specific multiscale color space embedding, the experiments are also performed over a single modality based on different scales of color spaces, and results are illustrated in Table 1. It can be observed from Table 1 that when the Fundus image is utilized and transformed into YCbCr color space at multiscale, it achieves 0.910 Accuracy for the AMD dataset compared to other single modalities with one color space transformation. However, the experimental performance is similar for single modality Fundus images when distinct color spaces are fused together. On the other hand, when modality-specific multiscale color space embedding strategy is assessed, then the performance of the MCGAEc model is elevated and achieved 0.947 Accuracy for the AMD dataset. Further, we have performed a comparative study demonstrated in Table 2 and 3, which shows the efficacy of the proposed MCGAEc method over the SOTA methods. The proposed study can be integrated into the clinical setting to help ophthalmologists with the diagnosis of the retinal disorder and, based on the observation, can predict the retinal disease.

4 Conclusion

We have presented a multi-modality MCGAEc deep learning model that assimilates the feature representations in various color spaces and emphasizes the vital discriminative features by multiscale mechanism. Besides, the global feature representation is incorporated into the proposed model by employing the attention mechanism at each path of the ensemble classifier. To analyze the behavior of the proposed model, extensive experiments were accomplished over the publicly available multi-modality AMD dataset and compared with the existing approach, which indicates the effectiveness of the proposed model. Additionally, the significance of the proposed method has been verified by performing experiments on a single modality with distinct color spaces, and the proposed MCGAEc model achieves higher evaluation measures compared to the considered cases. Moreover, we successfully incorporated ROI-specific mechanisms to learn essential features from the multi-modality imaging techniques used to diagnose retinal disorders and localize the affected region.

Acknowledgement. This work is supported by the Grant of SERB, Government of India (Ref. No: SPG/2022/000045) and IIT Kharagpur AI4ICPS I Hub Foundation (Ref No: IIT/SRIC/MM/ZDP/2023-2024/156).

Declarations

Conflict of interest. The authors declare that they have no conflict of interests.

References

1. Ali, M.A., Hossain, M.S., Hossain, M.K., Sikder, S.S., Khushbu, S.A., Islam, M.: AMDNet23: hybrid CNN-LSTM deep learning approach with enhanced preprocessing for age-related macular degeneration (AMD) detection. *Intell. Syst. Appl.* **21**, 200334 (2024)
2. Auw-Haendrich, C., Staubach, F., Witschel, H.: Optic disk drusen. *Surv. Ophthalmol.* **47**(6), 515–532 (2002)
3. Bird, A.C., et al.: An international classification and grading system for age-related maculopathy and age-related macular degeneration. *Surv. Ophthalmol.* **39**(5), 367–374 (1995)
4. Burlina, P.M., Joshi, N., Pacheco, K.D., Liu, T.A., Bressler, N.M.: Assessment of deep generative models for high-resolution synthetic retinal image generation of age-related macular degeneration. *JAMA Ophthalmol.* **137**(3), 258–264 (2019)
5. Castillo, M.M., et al.: Optical coherence tomography for the monitoring of neovascular age-related macular degeneration: a systematic review. *Ophthalmology* **122**(2), 399–406 (2015)
6. Chandra, M.A., Bedi, S.: Survey on SVM and their application in image classification. *Int. J. Inf. Technol.* **13**(5), 1–11 (2021)
7. Dai, Y., Gao, Y., Liu, F.: TransMED: transformers advance multi-modal medical image classification. *Diagnostics* **11**(8), 1384 (2021)
8. Das, D., Nayak, D.R., Pachori, R.B.: CA-Net: a novel cascaded attention-based network for multi-stage glaucoma classification using Fundus images. *IEEE Trans. Instrum. Measur.* **72**, 2531110 (2023)
9. Das, V., Dandapat, S., Bora, P.K.: A data-efficient approach for automated classification of OCT images using generative adversarial network. *IEEE Sens. Lett.* **4**(1), 1–4 (2020)
10. El-Den, N.N., et al.: Scale-adaptive model for detection and grading of age-related macular degeneration from color retinal Fundus images. *Sci. Rep.* **13**(1), 9590 (2023)
11. Fang, H., Shang, F., Fu, H., Li, F., Zhang, X., Xu, Y.: Multi-modality images analysis: a baseline for glaucoma grading via deep learning. In: Fu, H., Garvin, M.K., MacGillivray, T., Xu, Y., Zheng, Y. (eds.) OMIA 2021. LNCS, vol. 12970, pp. 139–147. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87000-3_15
12. Ferris III, F.L., et al.: Clinical classification of age-related macular degeneration. *Ophthalmology* **120**(4), 844–851 (2013)
13. Fu, H., et al.: Disc-aware ensemble network for glaucoma screening from Fundus image. *IEEE Trans. Med. Imaging* **37**(11), 2493–2501 (2018)
14. Green, W.R., Wilson, D.J.: Choroidal neovascularization. *Ophthalmology* **93**(9), 1169–1176 (1986)
15. Iqbal, K., Odetayo, M., James, A., Salam, R.A., Talib, A.Z.H.: Enhancing the low quality images using unsupervised colour correction method. In: 2010 IEEE International Conference on Systems, Man and Cybernetics, pp. 1703–1709. IEEE (2010)
16. Jemshi, K.M., Gopi, V.P., Issac Niwas, S.: Development of an efficient algorithm for the detection of macular edema from optical coherence tomography images. *Int. J. Comput. Assist. Radiol. Surg.* **13**(9), 1369–1377 (2018). <https://doi.org/10.1007/s11548-018-1795-6>
17. Karri, S.P.K., Chakraborty, D., Chatterjee, J.: Transfer learning based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration. *Biomed. Opt. Express* **8**(2), 579–592 (2017)

18. Kruper, J., et al.: Convolutional neural network-based classification of glaucoma using optic radiation tissue properties. *Commun. Med.* **4**(1), 72 (2024)
19. Kulkarni, V.Y., Sinha, P.K.: Random forest classifiers: a survey and future research directions. *Int. J. Adv. Comput* **36**(1), 1144–1153 (2013)
20. Lam, C., et al.: Performance of artificial intelligence in detecting diabetic macular edema from Fundus photography and optical coherence tomography images: a systematic review and meta-analysis. *Diabetes Care* **47**(2), 304–319 (2024)
21. Lee, C.S., Baughman, D.M., Lee, A.Y.: Deep learning is effective for classifying normal versus age-related macular degeneration oct images. *Ophthalmol. Retina* **1**(4), 322–327 (2017)
22. Li, A., Cheng, J., Wong, D.W.K., Liu, J.: Integrating holistic and local deep features for glaucoma classification. In: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1328–1331. IEEE (2016)
23. Li, X., et al.: Multi-modal multi-instance learning for retinal disease recognition. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 2474–2482 (2021)
24. Liu, Y.Y., Chen, M., Ishikawa, H., Wollstein, G., Schuman, J.S., Rehg, J.M.: Automated macular pathology diagnosis in retinal oct images using multi-scale spatial pyramid and local binary patterns in texture and shape encoding. *Med. Image Anal.* **15**(5), 748–759 (2011)
25. Mandal, S., Kumar, A., Chatterjee, J., Manjunatha, M., Ray, A.K.: Segmentation of blood smear images using normalized cuts for detection of malarial parasites. In: 2010 Annual IEEE India Conference (INDICON), pp. 1–4. IEEE (2010)
26. Mokwa, N.F., Ristau, T., Keane, P.A., Kirchhof, B., Sadda, S.R., Liakopoulos, S., et al.: Grading of age-related macular degeneration: comparison between color Fundus photography, fluorescein angiography, and spectral domain optical coherence tomography. *J. Ophthalmol.* **2013**, 385915 (2013)
27. Morano, J., Hervella, Á.S., Rouco, J., Novo, J., Fernández-Vigo, J.I., Ortega, M.: Weakly-supervised detection of AMD-related lesions in color Fundus images using explainable deep learning. *Comput. Methods Programs Biomed.* **229**, 107296 (2023)
28. Naik, S.K., Murthy, C.: Hue-preserving color image enhancement without gamut problem. *IEEE Trans. Image Process.* **12**(12), 1591–1598 (2003)
29. Peng, Y., et al.: DeepSeeNet: a deep learning model for automated classification of patient-based age-related macular degeneration severity from color Fundus photographs. *Ophthalmology* **126**(4), 565–575 (2019)
30. Ryan, S.J., Mittl, R.N., Maumenee, A.E.: The disciform response: an historical perspective. *Albrecht Von Graefes Arch. Klin. Exp. Ophthalmol.* **215**(1), 1–20 (1980)
31. Saine, P.J., Tyler, M.E.: *Ophthalmic photography: retinal photography, angiography, and electronic imaging*, vol. 132. Butterworth-Heinemann Boston (2002)
32. Schaal, K.B., Freund, K.B., Litts, K.M., Zhang, Y., Messinger, J.D., Curcio, C.A.: Outer retinal Tubulation in advanced age-related macular degeneration: optical coherence tomographic findings correspond to histology. *Retina* **35**(7), 1339–1350 (2015)
33. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. *arXiv preprint [arXiv:1803.02155](https://arxiv.org/abs/1803.02155)* (2018)
34. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *J. Big Data* **6**(1), 1–48 (2019)

35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
36. Sumengen, B., Manjunath, B.: Multi-scale edge detection and image segmentation. In: 2005 13th European Signal Processing Conference, pp. 1–4. IEEE (2005)
37. Sun, Y., Li, S., Sun, Z.: Fully automated macular pathology detection in retina optical coherence tomography images using sparse coding and dictionary learning. *J. Biomed. Opt.* **22**(1), 016012–016012 (2017)
38. Tak, N., Reddy, A.J., Martel, J., Martel, J.B.: Clinical wide-field retinal image deep learning classification of exudative and non-exudative age-related macular degeneration. *Cureus* **13**(8), e17579 (2021)
39. van Velthoven, M.E., Faber, D.J., Verbraak, F.D., van Leeuwen, T.G., de Smet, M.D.: Recent developments in optical coherence tomography for imaging the retina. *Prog. Retin. Eye Res.* **26**(1), 57–77 (2007)
40. Wang, W., et al.: Learning two-stream CNN for multi-modal age-related macular degeneration categorization. *IEEE J. Biomed. Health Inform.* **26**(8), 4111–4122 (2022)
41. Wilde, C., Patel, M., Lakshmanan, A., Amankwah, R., Dhar-Munshi, S., Amoaku, W.: The diagnostic accuracy of spectral-domain optical coherence tomography for neovascular age-related macular degeneration: a comparison with Fundus fluorescein angiography. *Eye* **29**(5), 602–610 (2015)
42. Xing, X., Liang, G., Zhang, Y., Khanal, S., Lin, A.L., Jacobs, N.: Advit: vision transformer on multi-modality pet images for Alzheimer disease diagnosis. In: 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI), pp. 1–4. IEEE (2022)
43. Yoo, T.K., Choi, J.Y., Seo, J.G., Ramasubramanian, B., Selvaperumal, S., Kim, D.W.: The possibility of the combination of oct and Fundus images for improving the diagnostic accuracy of deep learning for age-related macular degeneration: a preliminary experiment. *Med. Biol. Eng. Comput.* **57**, 677–687 (2019)
44. Zhang, Y., Yan, C., Yang, Z., Zhou, M., Sun, J.: Multi-omics deep-learning prediction of homologous recombination deficiency-like phenotype improved risk stratification and guided therapeutic decisions in gynecological cancers. *IEEE Journal of Biomedical and Health Informatics* (2023)



PolSAR Image Classification Using Superpixel Profile and CNN

Nabajyoti Das^(✉), Swarnajyoti Patra, and Amos Bortiew

Tezpur University, Tezpur 784028, Assam, India
nabajd@tezu.ernet.in

Abstract. In recent years, Convolutional Neural Network (CNN) based frameworks are being applied to polarimetric synthetic aperture radar (PolSAR) image classifications and achieved improved results. However, the performance of CNN for PolSAR image classification is greatly dependent upon the selection of polarimetric features. Although CNN automatically extracts abstract high level features from the data, it is still beneficial to incorporate additional hand-crafted features to enhance the classification results. In this research, to incorporate spatial information of the pixels in classification process, first, a profile of the PolSAR image is constructed by using superpixel algorithm. Then, the constructed superpixel profile is fed into a CNN model for classification. The experiment conducted on three real PolSAR datasets highlights the utility of superpixel profiles. For all the three datasets, our proposed method demonstrates a consistent improvement of at least 3% in classification accuracy in comparison to the state-of-the-art CNN model.

Keywords: PolSAR Images · Superpixel Profile · Convolutional Neural Network (CNN)

1 Introduction

Recently, polarimetric synthetic aperture radar (PolSAR) imagery has become an increasingly important component in the process of monitoring the surface of the earth due to its application in a variety of fields, including classification of land cover and crops, monitoring of snow and glaciers, identification of ships *etc..* This is possible because of its ability to monitor the earth's surface in all weather conditions and work both during day as well as night.

The advancement of PolSAR image classification led to the introduction of a number of polarimetric-based feature extraction techniques, such as Cloude-Pottier decomposition [3], Freeman and Durden method [5], as well as the Yamaguchi four-component decomposition [20]. In addition to polarimetric features, traditional features like colour features, morphological profiles, textural features, as well as spatial and semantic object-oriented features are also utilized for PolSAR image classification [4, 15, 16, 18, 25]. In the literature, classifiers such as Wishart classifier, maximum likelihood classifier, decision tree and support vector machine are used for PolSAR image classification. Recently, deep learning algorithms have shown promising result.

Lin et al. [11] used simple CNN for land cover classification. Yang et al. [21] introduced a feature selection technique for classification of PolSAR image by using 1-D CNN. Si-Wei Chen and Chen-Song Tao [2] leveraged expert knowledge of target scattering and polarimetric features to improve CNN training and classification. Han et al. [6] performed PolSAR image classification using optimal feature selection and convolutional neural network (CNN). Hua et al. [7] introduced a multiscale sequential network with an attention mechanism for PolSAR image classification. Jamali et al. [8] used deep CNNs incorporated with the Haar wavelet transform for efficient feature extraction to enhance the classification accuracy of PolSAR images. The success of all these CNN models are dependent upon the quality of the polarimetric features they extract during the learning process.

Superpixel algorithms have been extensively used within the domain of computer vision to improve the performance of image segmentation and classification tasks. One of the advantages of using superpixels is that it can provide a more compact and efficient representation of the image data while preserving important spatial information. Thus, it can be useful to incorporate better spatial information for the classification of the PolSAR images. Liu et al. [12] utilized superpixels for the very first time and developed a superpixel-based classification framework for PolSAR images. Wang and Liu [19] used superpixel-level scattering mechanism (SM) for ship detection from the PolSAR images. Qin et al. [17] applied SLIC superpixel algorithm for PolSAR image segmentation. Most of the techniques in the literature exploit superpixel as a post-processing tool for improving the classification accuracy.

In this paper, a superpixel profile (SP) for the PolSAR image is constructed by applying the existing superpixel algorithm. The profile is constructed in such a way that it incorporates sufficient spatial information of the PolSAR image. Then, the constructed SP is utilized by a deep CNN model for classification. While CNNs are primarily designed for feature extraction and hierarchical representation learning from local patches, superpixel algorithm aims to capture global spatial information. Superpixel algorithm group pixels with similar characteristics into compact regions, preserving spatial context. Combining SP with CNNs allows us to leverage both local features extracted by CNNs and global spatial information from SP, resulting in improved classification accuracy for PolSAR images. The experiment conducted on three real PolSAR datasets shows the potentiality of the proposed superpixel profile. The remaining sections are organized as follows. Section 2 presents the proposed method. Section 3 provides the description of the datasets that are used for the experiment. The analysis of the experimental results is provided in Sect. 4. Finally, Sect. 5 wraps up the paper with conclusion.

2 Proposed Method

In this paper, we proposed a method by combining CNN with superpixels that utilises spatial information present in the PolSAR image. CNN was chosen for its established effectiveness in handling patch-based features. Its capability to learn hierarchical representations of local features aligns well with the output of

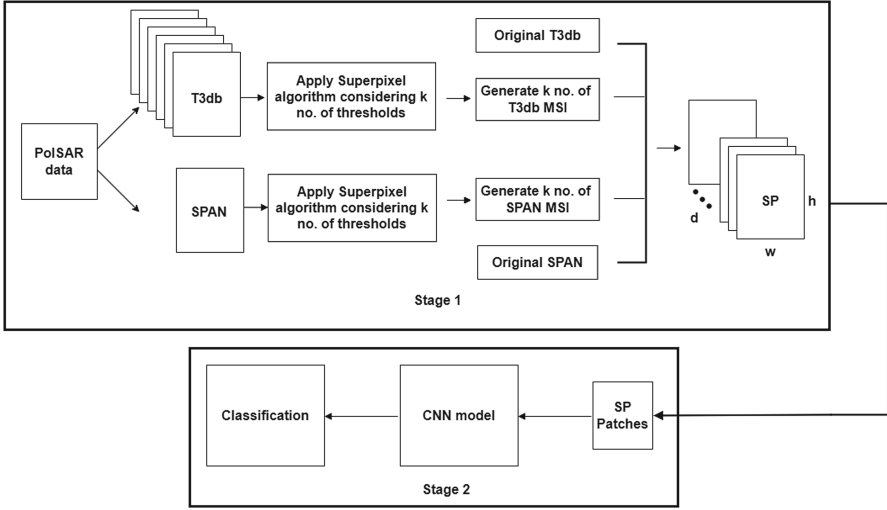


Fig. 1. Complete framework of the proposed method. Stage 1: Construction of SP from the PolSAR data utilizing T3db and SPAN. Stage 2: Classification using Deep CNN incorporating the constructed SP.

superpixel segmentation. Our method is implemented in two stages: In the first stage, superpixel profile is constructed by incorporating polarimetric features, and in the second stage, this handcrafted superpixel profile is fed into the CNN model for training and classification. By providing additional spatial information with the superpixel profile, our CNN model shows better ability to learn and classify the PolSAR images. The complete framework of our proposed method is shown in Fig. 1. The detailed explanation of the proposed method is given in the following sections.

2.1 Stage 1: Construction of the Superpixel Profile

In this stage, the spatial information present in the PolSAR image is utilized by exploiting superpixel algorithms. Although there are numerous superpixel algorithms available in the literature, in this work, three superpixel algorithms such as Simple Linear Iterative Clustering (SLIC) [1], Improved SLIC (iSLIC) [22], and Entropy Rate Superpixel (ERS) [13] are explored. These three algorithms are chosen based on their popularity, previous success in image processing tasks, and potential applicability to PolSAR images. SLIC is a widely used popular superpixel segmentation method that aims to generate superpixels with compact shapes and approximately uniform sizes. SLIC combines clustering and distance measures in a compact optimization framework to achieve efficient and accurate segmentation results. iSLIC is an improved version of SLIC, which uses the entire bands of the image, instead of few principal components. ERS is a graph-based superpixel segmentation method widely used in computer vision.

It creates superpixels by dividing the image into compact, homogeneous, and size-balanced sub-graphs.

In our proposed approach we use superpixel algorithm as a feature extraction technique to extract homogeneous regions of the image to feed into the CNN model. In superpixel algorithm the neighbour pixels that provides similar information are grouped together called superpixels. Thus, superpixels represent the image in term of some homogeneous regions that provides spatial information of the image. In detail, by applying a superpixel algorithm considering a threshold value t , it partitions the pixels on the image into multiple superpixels, each containing similar neighbour pixels of the input image. After obtaining the superpixels, the intensity value of each superpixel is computed by taking the average intensity value of the pixels that belong to it. Then a Mean Superpixel Image (MSI) associated to the threshold value t is created by representing each pixel of the input image using its superpixel intensity value. Since superpixels formed by considering a single threshold value are not sufficient to provide adequate spatial information of the input image. In order to provide sufficient spatial information, in this work, multiple MSI are generated by varying the threshold value in a wide range. Then a superpixel profile (SP) is constructed by concatenating all the generated MSI along with SPAN and T3db features of the original image. In more detail, suppose I be the original input image and t_1, t_2, \dots, t_k are the considered threshold values, then the SP of image I is constructed as:

$$I_{SP} = \{I_{SPAN}, I_{T3db}, I_{SPAN_MSI}^{t_1}, \dots, I_{SPAN_MSI}^{t_k}, I_{T3db_MSI}^{t_1}, \dots, I_{T3db_MSI}^{t_k}\} \quad (1)$$

where,

I_{SPAN} and I_{T3db} represents the SPAN feature and six T3db features of the PolSAR image I , respectively.

$I_{SPAN_MSI}^{t_1}, \dots, I_{SPAN_MSI}^{t_k}$ and $I_{T3db_MSI}^{t_1}, \dots, I_{T3db_MSI}^{t_k}$ are the MSI generated from the SPAN and T3db features respectively by the superpixel algorithm considering thresholds t_1, t_2, \dots, t_k .

The constructed SP can be represented as a $h \times w \times d$ cube, where h stands for height of the cube, w stands for width and d is the dimension of the cube. The Stage 1 of Fig. 1 illustrates the construction of SP.

2.2 Stage 2: Classification with Deep CNN

After the superpixel profile (SP) is constructed, patches are created for the PolSAR image which include sufficient spatial information and these patches are then used as input for the CNN. The CNN model is then trained to classify each patch based on the features represented in the superpixel profile. The model is trained using a supervised learning approach, in which it is provided with labeled samples of the patches and the desired output class for each patch [2].

Architecture of the CNN Model. The architecture of the CNN model used in this work is shown in Fig. 2. The model consists of three convolutional layers,

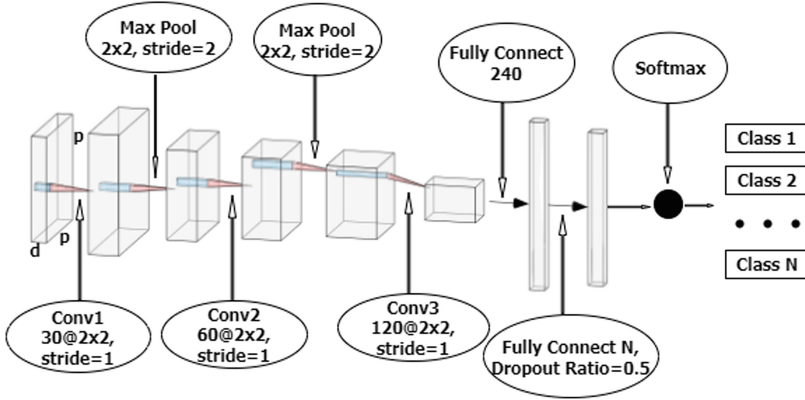


Fig. 2. CNN model architecture.

two max pooling layers, two fully connected layers, and a final softmax layer for classification. In greater detail, the three convolutional layers contains 30, 60, and 120 filters, with kernel size of 2×2 and stride of 1. The two max-pooling layers has size of 2×2 and stride of 2. Rectified linear units (ReLU) is used as the activation function for all the three convolutional layers as well as for the first fully connected layer. Furthermore, to reduce the problem of over-fitting, dropout with 0.5 ratio is applied to the second fully connected layer. From the Fig. 1 it can be seen that the initial input to the CNN model is the constructed SP. To leverage the spatial information by the CNN model, the constructed SP is fragmented into smaller patches of size $p \times p \times d$.

Finally, the frequently utilized stochastic gradient descent (SGD) coupled with back-propagation algorithms are employed to learn different weights and biases during the model's training process [10].

3 Description of Datasets

To validate the performance of our proposed method, we conduct experiments on three real-world PolSAR datasets commonly used in the field. For a comprehensive understanding of these datasets, their descriptions are provided below.

Flevoland AIRSAR Dataset. The Flevoland image, a fully polarimetric L-band SAR image with 10m spatial resolution, was acquired by NASA/JPL-Caltech AIRSAR. The image size is 750×1024 pixels, and its Pauli RGB representation is shown in Fig. 3(a). The ground truth map (Fig. 3(b)) contains 15 classes, including stembeans, peas, forest, lucerne, three types of wheat, beet, potatoes, bare soil, grass, rapeseed, barley, water, and buildings. The ground truth data is sourced from [23]. Table 1 lists the training and testing sample sizes.

Table 1. Train and Test Samples for Flevoland AIRSAR dataset

Sl. No.	Land Class	Training samples	Testing samples
1	Stem Beans	1000	6103
2	Peas	1000	9111
3	Forest	1000	14944
4	Lucerne	1000	9477
5	Wheat1	1000	17283
6	Beet	1000	10050
7	Potato	1000	15292
8	Bare Soil	1000	3078
9	Grass	1000	6269
10	Rapeseed	1000	12690
11	Barely	1000	7156
12	Wheat2	1000	10591
13	Wheat3	1000	21300
14	Water	1000	13476
15	Building	200	476

San-Francisco AIRSAR Dataset. The San Francisco image, a fully polarimetric L-band SAR image with 10 m spatial resolution, was acquired by NASA/JPL-Caltech AIRSAR. The image size is 900×1024 pixels, and its Pauli RGB representation is shown in Fig. 4(a). The ground truth map (Fig. 4(b)) con-

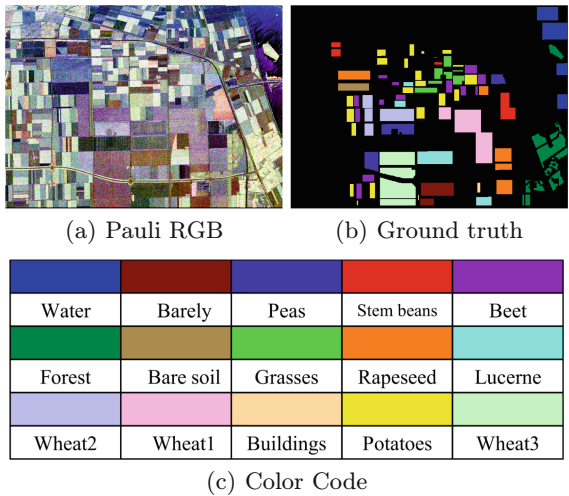


Fig. 3. Pauli RGB, ground truth and color code of Flevoland AIRSAR

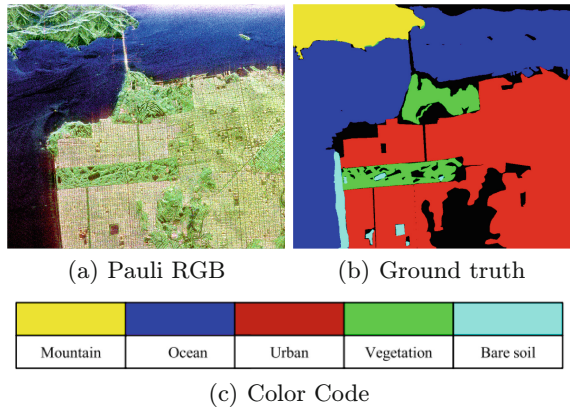


Fig. 4. Pauli RGB, ground truth and color code of San-Francisco AIRSAR

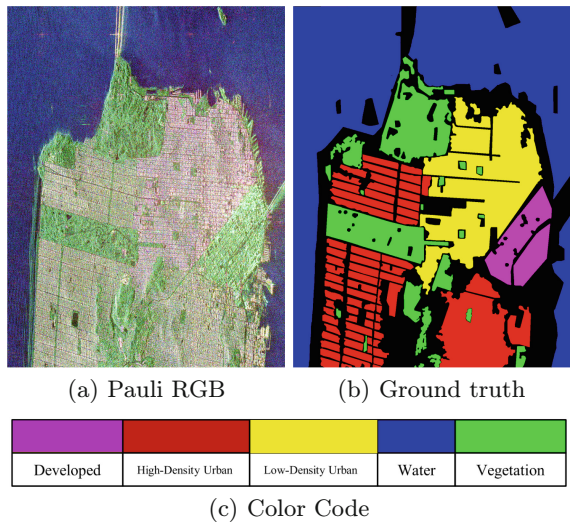


Fig. 5. Pauli RGB, ground truth and color code of San-Francisco RADARSAT-2

Table 2. Train and Test Samples for San-Francisco AIRSAR dataset

Sl. No.	Land Class	Training samples	Testing samples
1	Mountain	1000	13701
2	Ocean	1000	62731
3	Urban	1000	329566
4	Vegetation	1000	342795
5	Bare soil	1000	53509

Table 3. Train and Test Samples for San-Francisco RADARSAT-2 dataset

Sl. No.	Land Class	Training samples	Testing samples
1	Water	1000	852078
2	Vegetation	1000	237237
3	High-Density Urban	1000	351181
4	Low-Density Urban	1000	282975
5	Developed	1000	80616

Table 4. Classification accuracy on Flevoland AIRSAR dataset

Class	CNN	Wavelet-CNN	PolSARFormer	Proposed		
				ERS	iSLIC	SLIC
Stem beans	90.46	100	99.30	95.17	99.75	99.71
Peas	86.99	99.30	99.34	91.70	97.86	98.99
Forest	91.19	100	98.74	97.61	98.43	99.09
Lucerne	89.05	99.87	99.51	99.27	98.81	99.70
Wheat1	63.85	99.16	92.74	86.91	78.31	88.67
Beet	93.04	99.48	99.20	94.43	97.44	87.34
Potato	76.26	95.59	96.97	91.43	94.21	99.35
Bare Soil	99.90	98.64	100	99.90	99.94	99.87
Grass	73.58	90.83	96.84	93.19	95.65	84.67
Rapeseed	84.73	53.15	97.41	96.78	93.81	95.97
Barley	91.69	94.41	99.94	99.55	98.66	99.48
Wheat2	91.79	95.48	95.14	85.71	96.61	89.88
Wheat3	86.13	71.01	95.20	87.40	91.46	95.49
Water	99.69	0	0	99.94	99.95	99.87
Buildings	97.90	0	0	99.79	99.16	100
OA	85.53	81.87	88.46	93.21	94.41	95.44
AA	87.75	79.79	84.69	94.59	96.00	95.87
kappa	84.25	80.31	87.48	92.60	93.91	95.03

tains 5 classes: mountain, ocean, urban, vegetation, and bare soil. The ground truth data is sourced from [14]. The number of training and testing samples is provided in Table 2.

San-Francisco RADARSAT-2 Dataset. The second San Francisco image, a fully polarimetric C-band SAR image with 8m spatial resolution, was acquired by Canadian Space Agency RADARSAT-2. The image size is 1380×1800 pixels, and its Pauli RGB representation is shown in Fig. 5(a). The ground truth map (Fig. 5(b)) contains 5 classes: Water, Vegetation, High-Density Urban, Low-

Table 5. Classification accuracy on San-Francisco AIRSAR dataset

Class	CNN	Wavelet-CNN	PolSARFormer	Proposed		
				ERS	iSLIC	SLIC
Mountain	95.85	89.35	99.61	96.95	96.20	97.50
Water	94.57	54.95	93.68	96.58	96.91	92.73
Urban	97.23	77.99	97.85	97.95	97.32	96.95
Vegetation	85.69	92.03	96.78	89.17	90.83	92.62
Bare Soil	85.61	87.71	80.20	86.05	92.50	93.93
OA	91.29	84.50	94.39	93.28	94.17	94.58
AA	91.79	80.27	94.18	93.34	94.75	94.75
kappa	86.88	77.01	91.39	89.77	91.11	91.70

Table 6. Classification accuracy on San-Francisco RADARSAT-2 dataset

Class	CNN	Wavelet-CNN	PolSARFormer	Proposed		
				ERS	iSLIC	SLIC
Water	98.78	99.72	98.16	99.42	99.43	99.43
Vegetation	83.29	97.16	77.26	91.07	95.30	91.20
High Density Urban	81.63	82.07	89.38	80.18	90.39	89.78
Low Density Urban	82.72	98.02	92.02	94.07	85.32	95.34
Developed	89.02	85.02	94.79	93.81	87.44	93.28
OA	90.45	95.02	92.59	93.49	94.38	95.55
AA	87.09	92.40	90.32	91.71	91.58	93.81
kappa	86.32	92.86	89.40	90.67	91.92	93.61

Density Urban, and Developed. The ground truth data is sourced from [14]. For a detailed breakdown of the training and testing sets, please refer to Table 3.

4 Experimental Results

This section outlines the experimental setup and discusses the results and findings of the experiments.

4.1 Experimental Setting

To assess the effectiveness of the proposed method, we ran our experimental analysis on the three datasets as discussed above. In the experiment, the proposed method is compared with three models including a CNN model [24] and two state-of-the-art models, namely wavelet-CNN [8] and a ViT based PolSARFormer [9]. To show the potentiality of the proposed method irrespective of

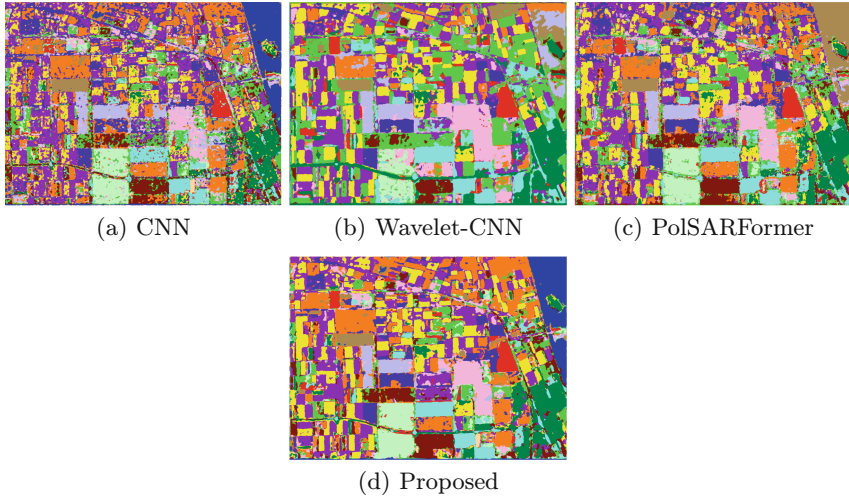


Fig. 6. Classification map of Flevoland AIRSAR data. (a) CNN (b) Wavelet-CNN (c) PolSARFormer (d) Proposed method using SLIC superpixel algorithm

the superpixel algorithms, Simple Linear Iterative Clustering (SLIC), Improved SLIC (iSLIC) and Entropy Rate Superpixel (ERS) are used.

The traditional polarimetric features are constructed from Lee-Speckle filtered T3db and SPAN Image, generated from the PolSAR data. We took four different threshold values for constructing four mean superpixel images(MSI) for each input image. The threshold values used are 20, 50, 100 and 200. Since T3db contains six features and SPAN contains a single feature which make it a sum of 7 original features. Hence, a total of $28(7 \times 4)$ superpixel feature images are obtained which are then stacked together with the original T3db and SPAN which finally makes it a total of 35 features. The extracted feature set is then utilized to create patches which are then fed into the proposed CNN model for classification. The patch size p for the proposed CNN model is taken as 7 for all the three datasets. Hence, the size of the input SP patches becomes $7 \times 7 \times 35$. In the experiments, the model is trained for 300 epochs with learning rate of 0.0001 via SGD with 64 mini-batch.

PolSAR image preprocessing, T3 matrix generation, SPAN image creation, and T3-to-T3db conversion were performed using PolSARpro v6.0.2. Superpixel (SP) construction was done in MATLAB, and the CNN model was implemented in Python using TensorFlow.

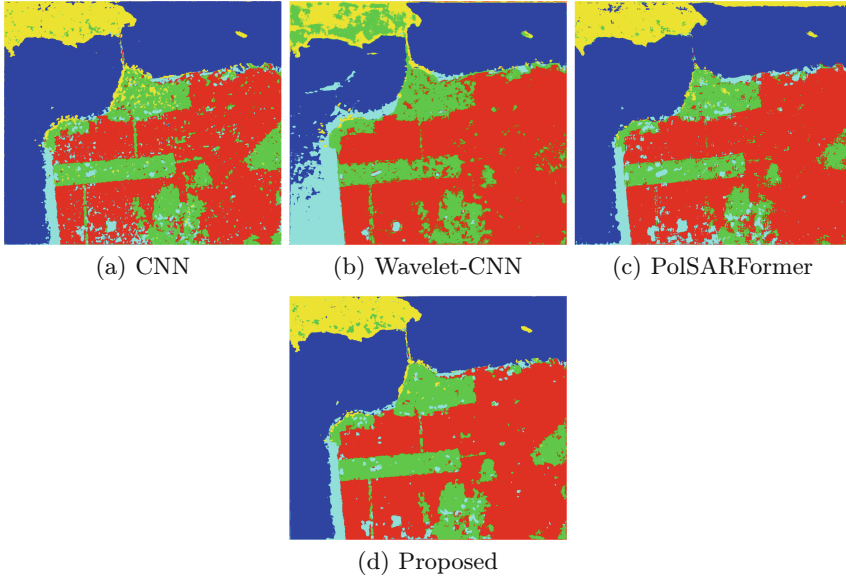


Fig. 7. Classification map of San-Francisco AIRSAR data. (a) CNN (b) Wavelet-CNN (c) PolSARFormer (d) Proposed method using SLIC superpixel algorithm

4.2 Results

The classification accuracy of all methods is measured using class-wise accuracy, overall accuracy (OA), average accuracy (AA), and the kappa coefficient (kappa), which provide a robust assessment of the performance. The classification results over the Flevoland AIRSAR dataset, San-Francisco AIRSAR dataset and San-Francisco RADARSAT-2 dataset, are shown in Table 4, 5 and 6 respectively. The reported results show that the proposed method with SLIC algorithm performed significantly better than other methods, including that of the Wavelet-CNN and PolSARFormer which are considered state-of-the-art methods for PolSAR Image classification. For the Flevoland AIRSAR dataset, it is seen from Table 4 that the OA achieved without using SP i.e. for the existing literature based CNN model was 85.53% and the OA improved to a high of 95.44% which is an increase of almost 10% when integrating with SP i.e. our proposed method. Similarly, for the San-Francisco AIRSAR dataset, the OA improved to a maximum of 94.58% from 91.29%, as shown in Table 5. Also, for the San-Francisco RADARSAT-2 dataset, the highest OA achieved by our proposed method is 95.55% whereas OA produced by CNN without SP is 90.45% as shown in Table 6. Hence, from the results obtained it is evident that our proposed method is able to provide better results than other methods, including CNN, Wavelet-CNN and PolSARFormer for all the three PolSAR datasets. For qualitative analysis Fig. 6, 7 and 8 show the classification maps produced by our proposed method in comparison with the CNN, Wavelet-CNN and PolSARFormer models for all the three considered

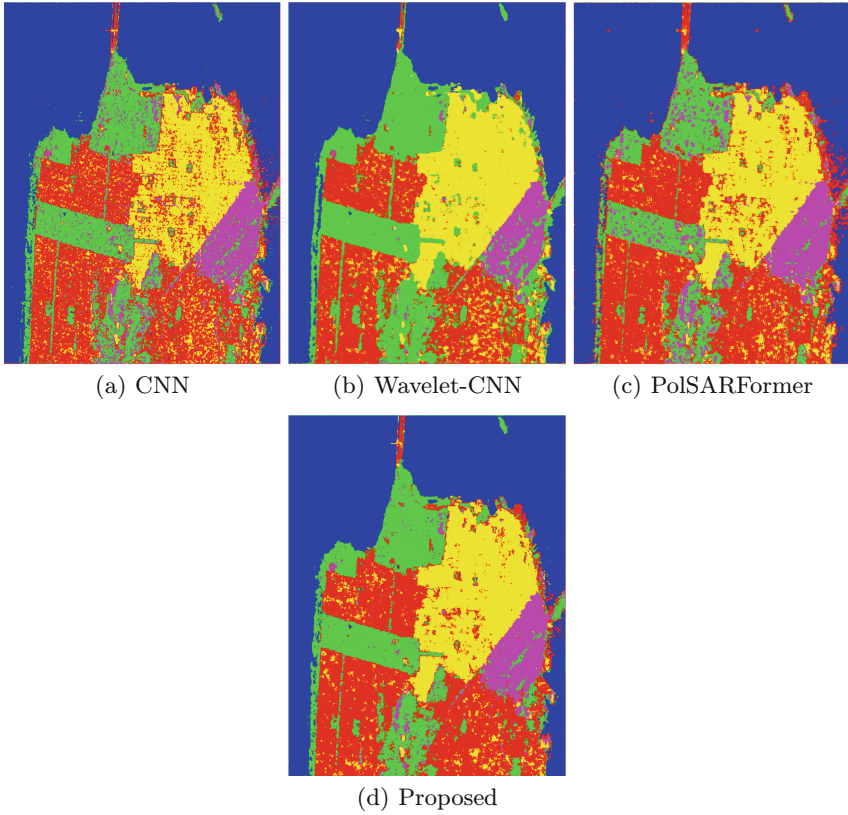


Fig. 8. Classification map of San-Francisco RADARSAT-2 data. (a) CNN (b) Wavelet-CNN (c) PolSARFormer (d) Proposed method using SLIC superpixel algorithm

dataset. It is seen that the proposed method can produce smoother classification maps with few misclassifications for all the three datasets.

4.3 Discussion

Based on the results of the experiments, it is clear that our approach of combining superpixel profile with deep CNN model is able to effectively incorporate spatial information for improving the classification performance of PolSAR images. It is evident that without the superpixel profile, the accuracy is significantly lower. This is because the superpixel profile is able to provide additional spatial information into the CNN model for better discrimination. This highlights the importance of incorporating spatial information in the classification process, which can be achieved through the use of superpixels. Furthermore, the results also show that among the three superpixel algorithms (SLIC, iSLIC, and ERS), SLIC provides the best OA for all three datasets. This further supports the effectiveness of the proposed approach in incorporating spatial information

to improve the classification performance of PolSAR images. It is noteworthy that in certain class instances, the CNN approach achieves better performance compared to at least one of the proposed methods. However, it still fails to provide consistent performance across all the classes. Similar inconsistencies were observed for the two other state-of-the-art methods, where it can be seen that in few classes, it performed exceptionally well but failed drastically for other classes. But, our proposed approach, while exhibiting slightly lower accuracy for few classes, has produced consistent performance across all the class instances. This consistency translates to a superior OA, which constitutes a key strength of our method. This desirable outcome of our proposed approach can be attributed to the combining of SP with CNN, which allows us to leverage both local features extracted by the CNN and global spatial information from SP, resulting in improved classification accuracy for PolSAR images.

5 Conclusion

This paper proposed a novel approach to improve the accuracy and efficiency of PolSAR image classification by combining a convolutional neural network with superpixels. The proposed method divided into two stages, where the first stage constructs the superpixel profile by incorporating spatial information extracted from the polarimetric features, and in the second stage, the CNN model is trained to classify each patch based on the features represented in the superpixel profile along with polarimetric features. The results clearly show that incorporating superpixel profile consistently achieves the highest overall accuracy (OA), average accuracy (AA), and kappa coefficient for all three datasets. This research highlights the potential of using superpixels as a feature extraction technique to extract spatial information by constructing superpixel profile, which can be fed into the CNN model to improve classification performance. This approach can be extended to other remote sensing data.

Acknowledgements. This research is partially supported by the Science and Engineering Research Board (SERB), under the Department of Science and Technology, Government of India, through Grant No. CRG/2020/003018.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(11), 2274–2282 (2012). <https://doi.org/10.1109/TPAMI.2012.120>
2. Chen, S.W., Tao, C.S.: PolSAR image classification using polarimetric-feature-driven deep convolutional neural network. *IEEE Geosci. Remote Sens. Lett.* **15**(4), 627–631 (2018). <https://doi.org/10.1109/LGRS.2018.2799877>
3. Cloude, S., Pottier, E.: An entropy based classification scheme for land applications of polarimetric SAR. *IEEE Trans. Geosci. Remote Sens.* **35**(1), 68–78 (1997). <https://doi.org/10.1109/36.551935>

4. Das, N., Pradhan, K., Patra, S.: Classification of polarimetric SAR image using JS-divergence profile. In: 2022 IEEE Calcutta Conference (CALCON), pp. 20–24 (2022). <https://doi.org/10.1109/CALCON56258.2022.10060487>
5. Freeman, A., Durden, S.: A three-component scattering model for polarimetric SAR data. *IEEE Trans. Geosci. Remote Sens.* **36**(3), 963–973 (1998). <https://doi.org/10.1109/36.673687>
6. Han, P., Chen, Z., Wan, Y., Cheng, Z.: PoLSAR image classification based on optimal feature and convolution neural network. In: IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, pp. 1735–1738 (2020). <https://doi.org/10.1109/IGARSS39084.2020.9324670>
7. Hua, W., Wang, X., Zhang, C., Jin, X.: Attention-based multiscale sequential network for PoLSAR image classification. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022). <https://doi.org/10.1109/LGRS.2022.3164464>
8. Jamali, A., Mahdianpari, M., Mohammadimanesh, F., Bhattacharya, A., Homayouni, S.: PolSAR image classification based on deep convolutional neural networks using wavelet transformation. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022). <https://doi.org/10.1109/LGRS.2022.3185118>
9. Jamali, A., Roy, S.K., Bhattacharya, A., Ghamisi, P.: Local window attention transformer for polarimetric SAR image classification. *IEEE Geosci. Remote Sens. Lett.* **20**, 1–5 (2023). <https://doi.org/10.1109/LGRS.2023.3239263>
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017). <https://doi.org/10.1145/3065386>
11. Lin, W., Liao, X., Deng, J., Liu, Y.: Land cover classification of RADARSAT-2 SAR data using convolutional neural network. *Wuhan Univ. J. Nat. Sci.* **21**(2), 151–158 (2016). <https://doi.org/10.1007/s11859-016-1152-y>
12. Liu, B., Hu, H., Wang, H., Wang, K., Liu, X., Yu, W.: Superpixel-based classification of polarimetric synthetic aperture radar images. In: 2011 IEEE RadarCon (RADAR), pp. 606–611 (2011). <https://doi.org/10.1109/RADAR.2011.5960609>
13. Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy rate superpixel segmentation. In: CVPR 2011, pp. 2097–2104 (2011). <https://doi.org/10.1109/CVPR.2011.5995323>
14. Liu, X., Jiao, L., Liu, F., Hou, X., Zhang, D., Tang, X.: PolSF: PolSAR image dataset on San Francisco. *arXiv preprint arXiv:1912.07259* (2019)
15. Marpu, P.R., Chen, K.S., Chu, C.Y., Benediktsson, J.A.: Spectral-spatial classification of polarimetric SAR data using morphological profiles. In: 2011 3rd International Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), pp. 1–3 (2011)
16. Masjedi, A., Valadan Zoej, M.J., Maghsoudi, Y.: Classification of polarimetric SAR images based on modeling contextual information and using texture features. *IEEE Trans. Geosci. Remote Sens.* **54**(2), 932–943 (2016). <https://doi.org/10.1109/TGRS.2015.2469691>
17. Qin, F., Guo, J., Lang, F.: Superpixel segmentation for polarimetric SAR imagery using local iterative clustering. *IEEE Geosci. Remote Sens. Lett.* **12**(1), 13–17 (2015). <https://doi.org/10.1109/LGRS.2014.2322960>
18. Uhlmann, S., Kiranyaz, S.: Integrating color features in polarimetric SAR image classification. *IEEE Trans. Geosci. Remote Sens.* **52**(4), 2197–2216 (2014). <https://doi.org/10.1109/TGRS.2013.2258675>
19. Wang, Y., Liu, H.: PolSAR ship detection based on superpixel-level scattering mechanism distribution features. *IEEE Geosci. Remote Sens. Lett.* **12**(8), 1780–1784 (2015). <https://doi.org/10.1109/LGRS.2015.2425873>

20. Yamaguchi, Y., Moriyama, T., Ishido, M., Yamada, H.: Four-component scattering model for polarimetric SAR image decomposition. *IEEE Trans. Geosci. Remote Sens.* **43**(8), 1699–1706 (2005). <https://doi.org/10.1109/TGRS.2005.852084>
21. Yang, C., Hou, B., Ren, B., Hu, Y., Jiao, L.: CNN-based polarimetric decomposition feature selection for PolSAR image classification. *IEEE Trans. Geosci. Remote Sens.* **57**(11), 8796–8812 (2019). <https://doi.org/10.1109/TGRS.2019.2922978>
22. Zhang, Y., Liu, K., Dong, Y., Wu, K., Hu, X.: Semisupervised classification based on SLIC segmentation for hyperspectral image. *IEEE Geosci. Remote Sens. Lett.* **17**(8), 1440–1444 (2020). <https://doi.org/10.1109/LGRS.2019.2945546>
23. Zhang, Z., Wang, H., Xu, F., Jin, Y.Q.: Complex-valued convolutional neural network and its application in polarimetric SAR image classification. *IEEE Trans. Geosci. Remote Sens.* **55**(12), 7177–7188 (2017). <https://doi.org/10.1109/TGRS.2017.2743222>
24. Zhou, Y., Wang, H., Xu, F., Jin, Y.: Polarimetric SAR image classification using deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **13**, 1935–1939 (2016)
25. Zou, B., Xu, X., Zhang, L.: Object-based classification of PolSAR images based on spatial and semantic features. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **13**, 609–619 (2020). <https://doi.org/10.1109/JSTARS.2020.2968966>



Know How Much Sensitive Precision and Recall Validity Measures Are?

Sumit Mishra^{1(✉)}, Srinibas Swain², and Ved Prakash²

¹ Department of Computer Science and Engineering, Rajiv Gandhi Institute of Petroleum Technology, Jais, Amethi, India

sumitm@rgipt.ac.in

² Department of Computer Science and Engineering, Indian Institute of Information Technology Guwahati, Guwahati, India

{srinibas,ved.prakash}@iiitg.ac.in

Abstract. For the performance evaluation of the clustering algorithm, evaluation metrics are used. For this purpose, the obtained set of clusters are compared with the actual set of clusters (or gold standard). Various evaluation metrics have been proposed in the past. One important question regarding these evaluation metrics is – how good are these metrics for evaluating the performance of the clustering algorithm? Wagner et al. have proposed some of the properties of these evaluation metrics. The evaluation metric should also have a high sensitivity value to capture the change in the clustering result/gold standard along with these properties. In this paper, we compute the sensitivity of two commonly used evaluation metrics – Precision and Recall. We also show that the sensitivity of Precision and Recall is polynomial with respect to the number of data-points.

Keywords: Sensitivity · Contingency table · Hasse diagram

1 Introduction

In the clustering domain, the evaluation metrics are used to evaluate the performance of the clustering algorithm. These metrics judge the quality of the obtained set of clusters against the actual set of clusters. Various evaluation metrics are used for evaluating the performance of the clustering results, like F-measure, Rand Index, Adjusted Rand Index, etc. One important question regarding these evaluation metrics is – how good are these metrics for evaluating the performance of the clustering algorithm? One algorithm's importance over the other is decided based on the value obtained by these evaluation metrics, so it is very important to be sure about our judging criteria (*i.e.*, evaluation metrics). Recently, a detailed review of F-measure has been discussed in [7]. Various important properties of these evaluation metrics have been suggested by Wagner

et al. [16]. Recently, some authors have explored one important property of the evaluation metrics [13, 17], which is the Sensitivity¹ and is defined as follows:

Definition 1. *Given an evaluation metric EM , the sensitivity of an evaluation metric EM is the possible number of unique values which EM can provide for all the possible clustering results.*

Consider the elevation metric *Precision*. Let us consider 3 data-points. The possible values of *Precision* considering all the clustering results obtained from 3 data-points is $\{0, 1, 1/3, 0/0\}$. Thus, the sensitivity of *Precision* is 4.

Along with satisfying the other properties, the evaluation metric should also have a high value of sensitivity. Now, the question is how to obtain the value of sensitivity for an evaluation metric. In this work, we are focusing on two popular validity measures – *Precision* and *Recall*. In our work, we will first discuss a naive approach to obtain sensitivity. Using this naive approach, we can obtain the sensitivity of all the validity measures. However, this approach is not practical as it explores all the possible clustering results (B_n for n data-points) and their combinations (B_n^2 for n data-points) to obtain the sensitivity, where B_n is the n^{th} Bell Number [1, 4, 10, 12]. We will discuss the number of possible clustering results and their combinations in detail in the next section. Next, some theoretical results are established, which helps in computing the sensitivity. We have also proved that the sensitivity of *Precision* and *Recall* is bounded by $\mathcal{O}(n^4)$ for n data-points. At the end, an approach to compute the sensitivity for *Precision* and *Recall* is presented using Hasse diagram². This approach is also exponential, but better than the naive approach. The bound on the sensitivity of *Precision* and *Recall* (i.e., $\mathcal{O}(n^4)$) gives us hope that some better algorithms can be developed in the future.

2 Preliminaries

Let $\mathbb{D} = \{p_1, p_2, \dots, p_n\}$ be the set of n data-points which can be grouped in different ways. Let $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$ and $\mathcal{C}' = \{C'_1, C'_2, \dots, C'_L\}$ be two such ways such that $\cup_{k=1}^K C_k = \cup_{l=1}^L C'_l = \mathbb{D}$ and $C_k \cap C_{k'} = C'_l \cap C'_{l'} = \phi$ for $1 \leq k \neq k' \leq K$ and $1 \leq l \neq l' \leq L$.

The cardinality of cluster C_k is n_{k*} , and the cardinality of cluster C'_l is n_{*l} . The number of data-points that are in both the clusters C_k and C'_l is n_{kl} . Let \mathcal{C} be the gold standard and \mathcal{C}' be the obtained clustering result. Two clustering results are compared using the notion of *contingency table*. The *contingency table* $M = (n_{kl})$ of the two clustering results \mathcal{C} and \mathcal{C}' is a $K \times L$ matrix. The kl^{th} entry of this matrix is obtained by the common number of data-points in C_k and C'_l . The contingency table for \mathcal{C} and \mathcal{C}' is shown in Table 1.

¹ Do not be confused with *Precision*, as some authors use the word sensitivity and *Precision* interchangeably.

² The readers are encouraged to read about the hasse diagram in [6].

Table 1. Contingency table for comparing two clustering result.

\mathcal{C}	\mathcal{C}'				Sums
	\mathcal{C}'_1	\mathcal{C}'_2	\dots	\mathcal{C}'_L	
\mathcal{C}_1	n_{11}	n_{12}	\dots	n_{1L}	n_{1*}
\mathcal{C}_2	n_{21}	n_{22}	\dots	n_{2L}	n_{2*}
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
\mathcal{C}_K	n_{K1}	n_{K2}	\dots	n_{KL}	n_{K*}
Sums	n_{*1}	n_{*2}	\dots	n_{*L}	$n_{**} = n$

The total number of clustering results for n data-points is obtained using n^{th} Bell number [4], B_n which satisfies the recurrence relation in Eq. (1) [4].

$$B_{n+1} = \sum_{k=0}^n \binom{n}{k} B_k, \quad B_0 = B_1 = 1 \quad (1)$$

The total number of clustering results is B_n . When two clustering results are compared, then one clustering result is the gold standard, and another is obtained by some clustering algorithm. Any of the B_n clustering results can be the gold standard as well as the obtained clustering result. Thus, there are a total of $B_n \times B_n$ ways in which two clustering results can be compared. Hence, the total number of contingency tables is also $B_n \times B_n$. The growth rate of the number of clustering results and their combinations *w.r.t.* the number of data-points is shown in Fig. 1. From this figure, it is clear that the number of clustering results and their combinations grows exponentially *w.r.t.* the number of data-points.

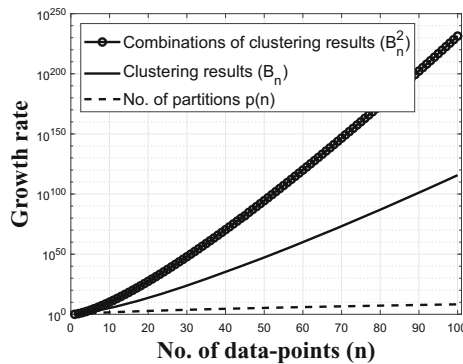


Fig. 1. Clustering results and their combinations.

True Positive (TP) is the number of pairs of data-points that are present in both \mathcal{C} and \mathcal{C}' . *False Positive* (FP) is the number of pairs of data-points that

are present in \mathcal{C}' but not in \mathcal{C} , whereas, *False Negative* (FN) is the number of pairs of data-points that are present in \mathcal{C} but not in \mathcal{C}' . The number of pairs of data-points that are not present in either \mathcal{C} or \mathcal{C}' is basically *True Negative* (TN). Let the total number of pairs be S which is equal to $\binom{n}{2}$ so $S = TP + FP + FN + TN$. As the maximum number of pair of data-points be $\binom{n}{2}$, thus, the maximum value of TP, FP, FN and TN is $\binom{n}{2}$. The values of TP, FP, FN and TN are computed from the contingency table using Eqs. (2), (3), (4) and (5) respectively.

$$TP = \sum_{k,l} \binom{n_{kl}}{2} \quad (2)$$

$$FP = \sum_l \binom{n_{*l}}{2} - TP \quad (3)$$

$$FN = \sum_k \binom{n_{k*}}{2} - TP \quad (4)$$

$$TN = \binom{n}{2} - (TP + FP + FN) \quad (5)$$

The values of *Precision* and *Recall* from the contingency table is calculated using Eqs. (6) and (7) respectively.

$$Precision = P = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = R = \frac{TP}{TP + FN} \quad (7)$$

3 Approach

In this section, we first discuss the naive approach and then the proposed approach. In the naive approach, we do the following –

- Explore all possible clustering results that can be obtained from n points. There are B_n number of such clustering results where B_n represents the Bell number [1, 4, 10, 12].
- Obtain all the possible contingency tables by comparing all the combinations of B_n clustering results. There are B_n^2 such comparisons between the clustering results.
- For each contingency table, compute the value of Precision.
- Obtain the unique value of Precision/Recall to get the value of sensitivity for Precision/Recall.

As we have seen in Fig. 1, B_n^2 grows exponentially, so this approach is not feasible even for a small value of n (for $n = 10$, the value of B_n^2 is more than 10^{10}). Thus, a better approach is required.

Theorem 1. *Given a contingency table T and its transpose T' . The value of Precision for T is the same as the value of Recall for T' , and the value of Recall for T is the same as the value of Precision for T' .*

Proof. The contingency table T is shown in Table 1. The transpose of T is shown in Table 2. The values of True Positive, False Positive and False Negative for contingency table T' is calculated using Eqs. (8), (9) and (10) respectively.

Table 2. Transpose of the contingency table shown in Table 1.

\mathcal{C}'	\mathcal{C}				Sums
	\mathcal{C}_1	\mathcal{C}_2	\dots	\mathcal{C}_K	
\mathcal{C}'_1	n_{11}	n_{21}	\dots	n_{K1}	n_{*1}
\mathcal{C}'_2	n_{12}	n_{22}	\dots	n_{K2}	n_{*2}
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
\mathcal{C}'_L	n_{1L}	n_{2L}	\dots	n_{KL}	n_{*L}
Sums	n_{1*}	n_{2*}	\dots	n_{K*}	$n_{**} = n$

$$\text{TP}' = \sum_{l,k} \binom{n_{lk}}{2} \quad (8)$$

$$\text{FP}' = \sum_k \binom{n_{k*}}{2} - \text{TP}' \quad (9)$$

$$\text{FN}' = \sum_l \binom{n_{*l}}{2} - \text{TP}' \quad (10)$$

True positive for T and T' is the same (see Eqs. (2) and (8)), *i.e.*, $\text{TP} = \text{TP}'$. False Positive for T is same as the False Negative for T' (see Eqs. (3) and (10)), *i.e.*, $\text{FP} = \text{FN}'$. Similarly, False Negative for T is same as the False Positive for T' (see Eqs. (4) and (9)), *i.e.*, $\text{FN} = \text{FP}'$. The values of *Precision* and *Recall* from the contingency table T' is calculated using Eqs. (11) and (12) respectively.

$$P' = \frac{\text{TP}'}{\text{TP}' + \text{FP}'} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

$$R' = \frac{\text{TP}'}{\text{TP}' + \text{FN}'} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (12)$$

From Eqs. (6) and (12), it is clear that the value of *Precision* for T is the value of *Recall* for T' . Similarly, From Eqs. (7) and (11), it is clear that the value of *Recall* for T is the value of *Precision* for T' .

Corollary 1. *The value of sensitivity for Precision and Recall is the same.*

Proof. The value of *Precision* for T is same as the value of *Recall* for T' and vice versa. So, all the possible values of *Precision* and *Recall* for a particular value of n is same. Hence, the value of sensitivity for *Precision* and *Recall* is same.

Theorem 2. *For n data-points, the value of sensitivity for Precision and Recall is bounded by $\mathcal{O}(n^4)$.*

Proof. *Precision* is of the form $\frac{p}{q}$, $p \leq q$ where p and q are non-negative integers. $p = \text{TP}$ and the maximum value of TP can be $\frac{n(n-1)}{2}$ as the maximum number of pairs of data-points is $\frac{n(n-1)}{2}$. $q = \text{TP} + \text{FP}$ and the maximum value of $\text{TP} + \text{FP}$ can also be $\frac{n(n-1)}{2}$. Thus, there is an upper bound on the values of p and q which is $\frac{n(n-1)}{2} = \mathcal{O}(n^2)$. Thus, the value of sensitivity for *Precision* and *Recall* is bounded by $\mathcal{O}(n^4)$.

An upper bound on the sensitivity of various validity measures can easily be obtained using their definitions. This paper mainly focuses on Precision and Recall, so we have only obtained the bound for these two measures.

3.1 Motivating Example

Precision is the ratio of TP to the sum of TP and FP. The values of TP and FP are calculated from the contingency table. The contingency table of size $R \times C$, $1 \leq R, C \leq n$ for n data-points is filled with one of the partitions of an integer n . The maximum cardinality of a partition is n , and the maximum number of elements in the contingency table can be n^2 . So actually, the contingency table is filled by the partitions of an integer n and zeros. From Eq. (2), it is clear that the value of TP is calculated by the elements in the contingency table. The value of $\text{TP} + \text{FP}$ is calculated by the column sum of the contingency table. Similar to the elements in the contingency table, the column-sum is also one of the partitions of an integer n . However, unlike the elements in the contingency table, which can contain zeros, the column-sum does not have zeros (as there is no row with all elements equal to 0). So the partitions of an integer have an important role in calculating TP and FP, and we exploit this role to compute the sensitivity. The integer partition have been widely used in other works also [3, 5, 9].

TP is calculated considering all the elements in a partition, which is used to fill the contingency table using Eq. (2). Similarly $\text{TP} + \text{FP}$ is computed from the column-sum of the contingency table using Eq. (3). We can obtain the value of sensitivity if we can answer the following question –

“for a given column-sum, what are the possible partitions that can fill the contingency table?”

The answer to this question enables us to find all the values of Precision for a given column-sum, and we can ask this question for all the column-sum. Our work is motivated by providing an answer to this question. To answer this question, we are creating a graph considering all the partitions of n . This graph

is termed as *Hasse Diagram* [6] or *Poset of Integer Partitions* [14]. Each node in this graph is considered as column-sum, and all the reachable nodes from a particular node will give the partitions used to fill the contingency table with the given column-sum. Next, this graph is discussed.

3.2 Hasse Diagram

Here we discuss the hasse diagram in detail. In particular we focus on the definition, the process to generate the hasse diagram and the complexity of generating this hasse diagram.

Definition 2 (Hasse Diagram). *Let \mathcal{P}_n be the set of all partitions of a positive integer n . We can organize \mathcal{P}_n using a graphical representation called a Hasse diagram \mathbb{G}_n in the following way. We create the graph \mathbb{G}_n with vertices $V = \{u : u \in \mathcal{P}_n\}$ arranged in rows, according to rank. We then add edges $E = \{(u, v) : u, v \in \mathcal{P}_n \text{ and two summands of } v \text{ can be added to form } u\}$*

We represent a node in a Hasse diagram \mathbb{G}_n as **Node** = $\langle \text{nodeId}, \text{partition} \rangle$.

- **nodeId**: It corresponds to the unique id assigned to each node. It is represented as **Node.nodeId**.
- **partition**: It represents the partition of an integer n corresponding to the node. It is denoted as **Node.partition**.

Let $p(n)$ be the number of partitions of n and $p(n, K)$ be the number of partitions of rank K , so $p(n) = \sum_{K=1}^n p(n, K)$. The number of nodes in Hasse diagram \mathbb{G}_n is $p(n)$. The Hasse diagram \mathbb{G}_7 is shown in Fig. 2. The process to generate the Hasse diagram for an integer n is summarized in Algorithm 1.

Each partition gives the value of TP and TP+FP. When the partition is treated as the column-sum in the contingency table, then the value of TP+FP is obtained (From Eq. (3)). When the partition is used to fill the contingency table, then the value of TP is obtained (From Eq. (2)). Thus, Eqs. (2) and (3) need to be executed several times. To efficiently compute the value of TP and TP+FP during the course of our approach, we are performing $\frac{K(K-1)}{2}$ for each $1 \leq K \leq n$ in line 1 – 4 of Algorithm 1 and storing it for further use.

Adjacency list representation is used to store the graph. We have computed the size of the adjacency list $p(n)$ using Hardy-Ramanujan asymptotic partition formula [2] defined in Eq. (13).

$$p(n) \sim 1/4n\sqrt{3} \exp\left(\pi\sqrt{2n/3}\right) \quad (13)$$

The error rate between the exact number of partitions and the obtained number of partitions using Eq. (13) decreases as n increases [11]. Thus, there will be some space wastage. However, this formula's error rate is low, so space wastage will not be significant.

The Hasse diagram is generated in a rank-wise manner, starting from the nodes corresponding to partitions of rank 1 to rank n . There is only one partition

Algorithm 1. Hasse Diagram**Input:** n : An integer**Output:** \mathbb{G}_n : Hasse diagram from partitions of n

```

1:  $\text{combAr}[ ] \leftarrow \emptyset$  ▷ An array of size  $n$  to store the combinatorial
2: for  $k \leftarrow 1$  to  $n$  do
3:    $\text{combAr}[k] \leftarrow \binom{k}{2}$ 
4:  $L[ ] \leftarrow \emptyset$  ▷ An array of size  $p(n)$  to store the adjacency list of each node in the
   Hasse diagram
5:  $\text{id} \leftarrow 1$ 
6:  $\text{partition} \leftarrow \langle n \rangle$  ▷ Initial partition of rank 1
7:  $\text{comb} \leftarrow \binom{n}{2}$  ▷ Obtain TP (when  $n$  is used to fill the contingency table) and TP + FP
   (when  $n$  is the column-sum in the contingency table) corresponding to  $n$ 
8:  $\text{Node} \leftarrow \langle \text{id}, \text{partition} \rangle$  ▷ Create initial node
9:  $L[\text{id}].\text{node} \leftarrow \text{Node}$ 
10: Add  $\text{nodeId}$  to the Queue  $Q$ 
11:  $K_{\text{prev}} \leftarrow 0$  ▷ Rank of previously generated partition
12: while  $Q$  is not empty do
13:    $\text{currNodeId} \leftarrow \text{Dequeue from } Q$ 
14:    $\text{currNode} \leftarrow L[\text{currNodeId}]$  ▷ Get the node corresponding to  $\text{currNodeId}$ 
15:    $P \leftarrow \text{currNode.partition}$  ▷ Partition corresponding to  $\text{currNode}$  of the form
    $P = \langle \lambda_1, \lambda_2, \dots, \lambda_k \rangle$ 
16:    $K \leftarrow |P|$  ▷ Rank of partition  $P$ 
17:   if  $K = n$  then ▷ Rank of current partition is  $n$ 
18:     BREAK ▷ Hasse diagram is generated
19:   if  $K \neq K_{\text{prev}}$  then ▷ All the partitions of rank  $K_{\text{prev}}$  have been processed
20:     Re-initialize the Trie
21:   for  $k \leftarrow 1$  to  $K$  do ▷ Process each element of  $P$ 
22:     if  $\lambda_k \neq \lambda_{k-1}$  then ▷ Two consecutive elements of  $P$  are different
23:        $P'[1, \dots, K+1] \leftarrow \emptyset$  ▷ Initialize the partition of rank  $K+1$  which can be
   obtained from  $P$ 
24:       Copy the elements from  $P$  to  $P'$  except  $\lambda_k$ 
25:       for  $i \leftarrow 1$  to  $\lfloor \frac{\lambda_k}{2} \rfloor$  do
26:          $P'[K] \leftarrow i, P'[K+1] \leftarrow \lambda_k - i$  ▷ Last two elements of  $P'$ 
27:       Sort  $P'$  in non-increasing order of their elements
28:       if  $P'$  has already been generated using Trie then
29:          $\text{nodeId} \leftarrow \text{Id of the node corresponding to partition } P'$ 
30:          $L[\text{currNodeId}].\text{adjList} \cup \text{nodeId}$ 
31:       else ▷  $P'$  is newly generated
32:          $\text{id} \leftarrow \text{id} + 1$  ▷ Increment id for the next node
33:          $\text{comb} \leftarrow \text{FIND-COMBINATORIAL}(P', \text{combAr}[ ])$  ▷ Obtain TP
   (when  $P'$  is used to fill the contingency table) and TP + FP (when
    $P'$  is the column-sum in the contingency table) corresponding to
    $P'$ 
34:          $\text{Node} \leftarrow \langle \text{id}, P', \text{comb} \rangle$  ▷ Create new node
35:          $L[\text{currNodeId}].\text{adjList} \cup \text{id}$ 
36:         Insert  $P'$  and  $\text{id}$  into the Trie
37:         Enqueue  $\text{id}$  to  $Q$ 
38:    $K_{\text{prev}} \leftarrow K$ 

```

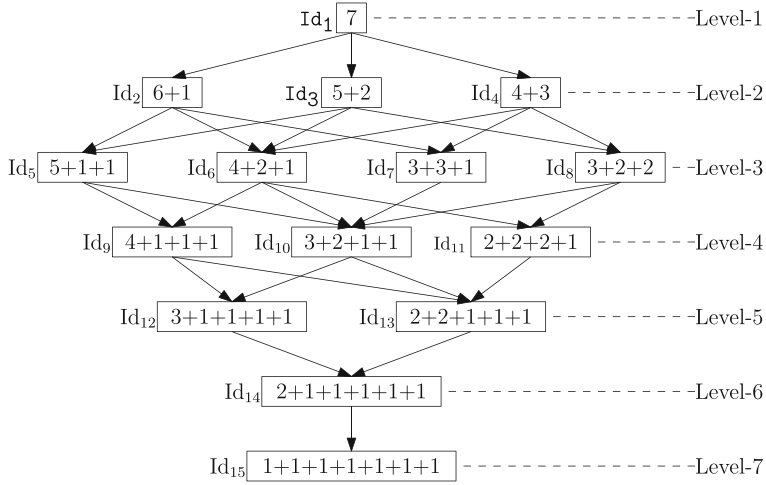


Fig. 2. Hasse Diagram for \mathcal{P}_7

of rank 1 which is $\langle n \rangle$. Initially, the node is created corresponding to this partition (Line 5 – 8). This node is inserted into the queue. In the process of inserting the node into the queue, we only insert its *id* to save space. With the help of *id* of a node, we can get the partition corresponding to this *id* in constant time by looking up in the adjacency-list (given in Line 4).

Now the *ids* of the nodes are dequeued from the queue to generate the new nodes. Once we have the *id*, we can get the partition corresponding to this *id*. If the rank of the partition corresponding to the dequeued *id* is equal to n , *i.e.*, all the partitions are generated, then the process to generate the Hasse diagram is terminated. Otherwise, the process of generating the Hasse diagram continues.

A partition $P = \langle \lambda_1, \lambda_2, \dots, \lambda_K \rangle$ of rank K can generate multiple partitions of rank $K+1$. Initially, the partition of rank $K+1$ (say P') is initialized to \emptyset (Line 23). The partition of rank $K+1$ is generated from each element λ_k , $1 \leq k \leq K$ of P . For this, all the elements of P except λ_k , are copied to P' (Line 24). The last two elements of P' are obtained by dividing λ_k in two parts (Line 26). An element λ_k generates $\lfloor \frac{\lambda_k}{2} \rfloor$ partitions of rank 2. In this algorithm, a partition element is divided *iff* the same partition element is not already divided into two parts. This is because if two partition elements are the same, their division into two parts yields the same new partitions.

Lemma 1. *Maximum number of partitions of rank $K+1$ generated from a partition P of rank K is $\mathcal{O}(n)$.*

Proof. Consider a partition $P = \langle \lambda_1, \lambda_2, \dots, \lambda_K \rangle$ of rank K . $\lambda_1 + \lambda_2 + \dots + \lambda_K = n$. The partitions of rank $K+1$ is generated from each element λ_k of P by dividing it into two parts. Thus, the total number of partitions of rank $K+1$ obtained from P is $\frac{\lambda_1}{2} + \frac{\lambda_2}{2} + \dots + \frac{\lambda_K}{2} = \frac{n}{2} = \mathcal{O}(n)$.

When the partitions are generated, it may be possible that the same partition of rank $K+1$ is generated from more than one partition of rank K . However, the same partition can be considered as a node only once. So when a partition is generated, we have to check whether it has already been generated or not. This can be performed using the Trie data structure [8, 15], which stores all the partitions of a particular rank at a time. The generated partition is sorted in non-increasing order of their elements, which in-turn helps in obtaining the partitions of rank $K+1$ from a partition of rank K and also check whether it has already been generated or not using Trie, efficiently. In Algorithm 1, when we obtain a partition P' of rank $K+1$, we perform the following –

- If the partition P' is already generated (Line 28), then we get the `nodeId` corresponding to P' (Line 29). Now the `nodeId` of P' is added to the adjacency-list of P as partition P' has been generated from P (Line 30).
- If the partition P' is not generated previously (Line 31), then a new node corresponding to this partition is created (Lines 32 – 34) and the `nodeId` of this newly created node is added to the adjacency-list of P (Line 35). The partition is also added to the Trie (Line 36). The `nodeId` of this newly generated node is added to the queue (Line 37).

Algorithm 2. FIND-COMBINATORIAL($P, \text{combAr}[\]$)

Input: A partition $P = \langle \lambda_1, \lambda_2, \dots, \lambda_K \rangle$ of rank K

Output: Combinatorial of partition P

```

1: combinatorial  $\leftarrow 0$ 
2: for  $k \leftarrow 1$  to  $K$  do
3:   if  $\lambda_k \geq 2$  then
4:     combinatorial  $\leftarrow$  combinatorial +  $\text{combAr}[\lambda_k]$ 
5: return combinatorial
```

When all the partitions of rank K are processed (*i.e.*, all the partitions of rank $K+1$ are generated), we re-initialize the Trie (Line 19 – 20) to store the partitions of rank $K+2$. In this way, we do not need to store all the partitions in the Trie. Thus, the space requirement reduces. When the partitions of rank K are stored in the form of a trie, then the time complexity to check whether a partition of rank K has been generated or not is $\mathcal{O}(K)$.

Complexity Analysis: We are performing $\frac{K(K-1)}{2}$ for each $1 \leq K \leq n$ in line 1–4 of Algorithm 1. An integer K is represented using $\mathcal{O}(\log K)$ bits. So performing $\frac{K(K-1)}{2}$ requires $\mathcal{O}(\log^2 K)$ time using grad school multiplication. Thus, the time complexity of lines 1–4 of Algorithm 1 is given as follows: $\sum_{K=1}^n \log^2 K = n \log^2 n$. The maximum number of partitions of rank $K+1$ obtained from a partition P of rank K is $\mathcal{O}(n)$ (from Lemma 1). The time complexity to generate a partition of rank $K+1$ from a partition of rank K is $\mathcal{O}(K+1)$. The generated partitions of rank $K+1$ are sorted in non-increasing order of their elements, which takes $\mathcal{O}(n)$ time

using counting sort. Each of these partitions is checked to see whether they have already been generated or not using Trie, which requires $\mathcal{O}(K+1)$ time. In case the partition has not already been generated, we need to add it to the Trie which requires $\mathcal{O}(K+1)$ time. For each generated partition of rank $K+1$, we obtain the possible value of TP or TP + FP in Line 33 using Algorithm 2. This step requires $\mathcal{O}(K+1)$ time. So the overall time complexity of Algorithm 1 is given by Eq. (14).

$$T_{\text{Hasse}} = n \log^2 n + \sum_{k=1}^K p(n, k) \cdot (n(K+1 + n + K+1 + K+1)) = n^2 p(n) \quad (14)$$

From a partition of rank K , we can get $\mathcal{O}(n)$ partitions of rank $K+1$ (From Lemma 1). Thus, there are $\mathcal{O}(n)$ outgoing edges from a node. Hence, the total number of edges in the Hasse Diagram is bounded by $\mathcal{O}(n \cdot p(n))$. As we are using adjacency list representation, the space required to store Hasse Diagram is $\mathcal{O}(p(n) + n \cdot p(n)) = \mathcal{O}(np(n))$. In our approach, the partitions are generated in a rank-wise manner. So there is no need to store all the partitions in the form of a Trie together. The Trie only stores the partitions of a particular rank $K+1$ when the partitions of rank K are processed. When all the partitions of rank $K+1$ are stored, then it requires $(K+1) \cdot p(n, K+1)$ space. Thus, the space required to store Trie is $\max_{1 \leq K \leq n} (K \cdot p(n, K))$.

4 Sensitivity Computation

The process to obtain the sensitivity of *Precision* is summarized in Algorithm 3. The importance of partitions in sensitivity computation is discussed in Sect. 3.1. We aim to obtain the list of all the nodes that are reachable from a given node in the Hasse Diagram, and this can be obtained by BFS, considering the given node as a source node. Any partition can be the column-sum, so this process must be repeated, considering each node as a source node while applying BFS traversal. The value obtained from the partition corresponding to the source node of BFS is the value of the denominator. In contrast, the value obtained from the partition corresponding to all the reachable nodes from the source node (including the source node itself) is the numerator's value. After obtaining all the possible values of the *Precision*, we compute the unique values and thus the sensitivity.

When we apply BFS to a node y (considering it as a source) and there is an edge (x, y) , then all the nodes reachable from y will also be reachable from x . It means when we have the list of reachable nodes from y , we can use that information while considering node x as a source. Thus, all the possible values of TP when BFS is applied to node y considering it as a source, will also be the value of TP for node x along with some other values. Now the question is how to use the already obtained TP values efficiently? The value of TP varies between 0 and $\frac{n(n-1)}{2}$ and we use this range to use the already obtained TP values efficiently.

To exploit this range information efficiently, we start applying BFS to the nodes at the last level (n^{th} level), then the nodes at the $n-1^{th}$ level, $n-2^{th}$ level, and so on. As TP varies between 0 and $\frac{n(n-1)}{2}$, so we create a boolean array

Algorithm 3. Obtain Sensitivity**Input:** \mathbb{G}_n : Hasse Diagram for an integer integer n **Output:** **sen**: Sensitivity of *Precision* for n points

```

1:  $S \leftarrow \binom{n}{2}$ 
2: sen_matrix[0, ..., S][0, ..., S]  $\leftarrow$  FALSE  $\triangleright$  Initialize a 2D boolean matrix of size
   ( $S + 1$ )  $\times$  ( $S + 1$ )
3:  $|V| \leftarrow$  Number of vertices in Hasse Diagram  $\mathbb{G}_n$ 
4: Node  $\leftarrow$  Last node in  $\mathbb{G}_n$ 
5: Node.TPAr[0, ..., S]  $\leftarrow$  FALSE  $\triangleright$  Boolean array of size  $S + 1$  for the last node
6: comb  $\leftarrow$  Node.comb  $\triangleright$  combinatorial for the last node is 0
7: Node.TPAr[comb]  $\leftarrow$  TRUE  $\triangleright$  True Positive value when BFS is applied to last node
   is 0
8: denom  $\leftarrow$  Node.comb  $\triangleright$  The value of denominator when Node.partition is the
   column-sum of the contingency table
9: sen_matrix[numer][denom]  $\leftarrow$  TRUE
10: for  $i \leftarrow |V| - 1$  to 1 do
11:   Node  $\leftarrow$  L[i]  $\triangleright i^{th}$  node in the Hasse diagram
12:   Node.TPAr[0, ..., S]  $\leftarrow$  FALSE  $\triangleright$  Boolean array of size  $S + 1$  for Node
13:   comb  $\leftarrow$  Node.comb
14:   Node.TPAr[comb]  $\leftarrow$  TRUE  $\triangleright$  The value of True Positive when Node.partition
   is used to fill contingency table
15:   for each node  $\in$  Node.adjList do
16:     Node.TPAr[ ]  $\leftarrow$  Node.TPAr[ ] || node.TPAr[ ]  $\triangleright$  Logical OR operation
       between the TPAr[ ] array of Node and its child node
17:   denom  $\leftarrow$  Node.comb  $\triangleright$  The value of denominator when Node.partition is the
   column-sum of the contingency table
18:   for  $j \leftarrow 0$  to S do
19:     if Node.TPAr[j] = TRUE then
20:       numer  $\leftarrow$  j
21:       Precision  $\leftarrow$   $\frac{\text{numer}}{\text{denom}}$ 
22:       Reduce Precision to its lowest form
23:       sen_matrix[numer][denom]  $\leftarrow$  TRUE
24: sen  $\leftarrow$  Number of TRUE in sen_matrix[ ][ ]
25: return sen

```

of size $\frac{n(n-1)}{2} + 1$ associated with each node of the Hasse diagram. This array stores the possible values of TP when BFS is applied to the associated node, considering that node as a source. The i^{th} entry in this array is TRUE if the TP value is i , otherwise, it is FALSE.

Let there be a node x at $K-1^{th}$ level and there be $\{\alpha_1, \alpha_2, \dots, \alpha_l\}$ nodes in the adjacency list of x . The possible values of TP when BFS is applied to node x is obtained by performing the OR operation among the boolean arrays (which stores TP values) of nodes $\{\alpha_1, \alpha_2, \dots, \alpha_l\}$. Along with all these obtained True Positive values, one more value obtained from the partition for node x is also considered.

Consider a node Id_6 from \mathbb{G}_7 shown in Fig. 2. The adjacency list of this node is $\text{Id}_6.\text{adjList} = \{\text{Id}_9, \text{Id}_{10}, \text{Id}_{11}\}$. When BFS is applied, considering node Id_9

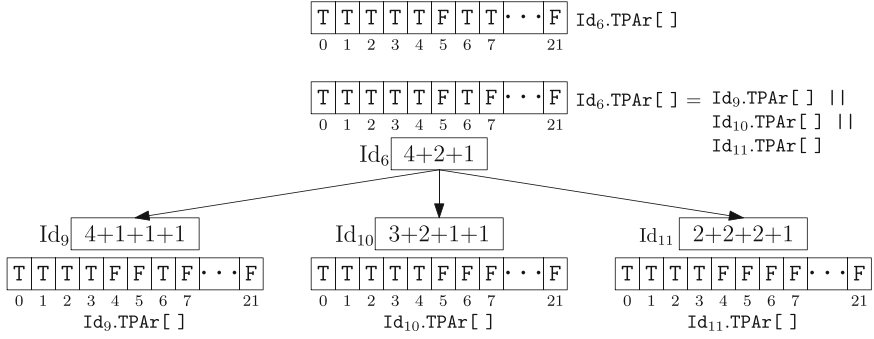


Fig. 3. Illustration of OR operation among the arrays of a node.

as a source, then the possible values of TP is $\{0, 1, 2, 3, 6\}$. The boolean array for the node Id_9 , i.e., $\text{Id}_9.\text{TPAr}[]$ is shown in Fig. 3. Similarly, the boolean array for the nodes Id_{10} and Id_{11} is shown in Fig. 3.

The possible values of TP for node Id_6 is obtained by performing OR operation among the three boolean arrays $\text{Id}_9.\text{TPAr}[]$, $\text{Id}_{10}.\text{TPAr}[]$ and $\text{Id}_{11}.\text{TPAr}[]$. Along with the obtained values of TP from OR operation, one more TP value obtained from the partition for node Id_6 is also considered.

Complexity Analysis: The maximum number of nodes in the adjacency list of any node at k^{th} level is $\mathcal{O}(n)$ (From Lemma 1). To obtain all the possible values of TP when BFS is applied to a node considering it as a source node, we need to perform OR operation among the TP boolean array of all the nodes in the adjacency list. As the size of the TP boolean array for each node is $\frac{n(n-1)}{2} + 1$, so one OR operation takes $\mathcal{O}(n^2)$ time. We need to perform $\mathcal{O}(n)$ such OR operations to obtain the TP values for a node at k^{th} level. Thus, the time complexity to obtain all the TP values for a node at k^{th} level is given by Eq. (15).

$$T_{\text{OR}} = \underbrace{n}_{\text{Size of adjacency list}} \times \underbrace{n^2}_{\text{OR operation}} = \mathcal{O}(n^3) \quad (15)$$

The size of the TP array associated with each node is $\mathcal{O}(n^2)$. So after BFS traversal considering a particular node as a source, we need to compute the value of Precision. There can be at-most $\mathcal{O}(n^2)$ such values. The value of *Precision* is of the form $\frac{p}{q}$, $p \leq q$ where p, q are non-negative integers and their maximum value is $\frac{n(n-1)}{2}$. To reduce the *Precision* to its lowest form, we first compute the gcd of p and q and then divide p and q by the gcd. The time complexity to find the gcd of p and q is $\mathcal{O}(\log \min(p, q)) = \mathcal{O}(\log n^2)$. The time to divide p and q by gcd is $\mathcal{O}(\log^2 q) = \mathcal{O}(\log^2 n^2)$. Thus, we can say that the time to reduce $\frac{p}{q}$ to its lowest form is no more than $\mathcal{O}(n)$. Hence, the time complexity to reduce all the Precision values obtained after applying BFS to a node is given by Eq. (16).

$$T_{\text{Pre}} = \underbrace{n^2}_{\text{Size of TP array}} \times \underbrace{n}_{\text{Lowest form convert}} = \mathcal{O}(n^3) \quad (16)$$

There are $p(n, K)$ number of nodes at the K^{th} level, and BFS is applied to all the nodes, considering them as the source. Thus, the time complexity to obtain the possible values of TP and TP+FP, and thus Precision, is obtained by Eq. (17).

$$T_{\text{sen}} = \sum_{K=n}^1 p(n, K) \cdot (T_{\text{OR}} + T_{\text{Pre}}) = 2n^3 \sum_{K=n}^1 p(n, K) = n^3 p(n) \quad (17)$$

The values of p and q vary between 0 and $\frac{n(n-1)}{2}$. So to obtain sensitivity, we create a boolean matrix of size $(S+1) \times (S+1)$ initialized by FALSE. For each p/q value of the Precision, the q^{th} column of the p^{th} row is set to TRUE. At the end, the count of TRUE in the matrix is the value of sensitivity. The time complexity of this technique is $\mathcal{O}(n^4)$. Thus, the overall time complexity to obtain the sensitivity is $\mathcal{O}(n^3 p(n))$. The sensitivity of Precision/Recall for $2 \leq n \leq 60$ data-points is shown in Fig. 4.

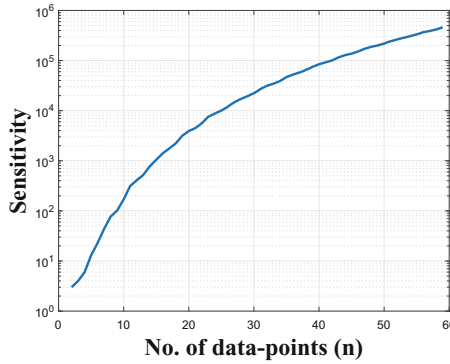


Fig. 4. The value of sensitivity.

An array of size $\mathcal{O}(n^2)$ is associated with each node, which requires $\mathcal{O}(n^2 p(n))$ space. To store all unique values of Precision, we create a 2D array of size $(S+1) \times (S+1)$ which requires $\mathcal{O}(n^4)$ space. The space complexity of Algorithm 1 is $\mathcal{O}(np(n))$. Thus, the overall space complexity of our approach is $\mathcal{O}(n^2 p(n))$.

The runtime of the proposed approach along with the naive approach is shown in Fig. 5. From this figure, it is clear that the time taken by our approach is less than the naive approach. As the time taken by the naive approach grows in the order of B_n^2 , we are unable to record the running time to calculate the sensitivity for $n \geq 10$, whereas using our approach one can record the running time. The reason for this is that the naive approach would explore all combinations of the clustering results which is $\mathcal{O}(B_n^2)$ and this is much bigger than the number of partitions $p(n)$ used in our approach. This shows the practical importance of our approach.

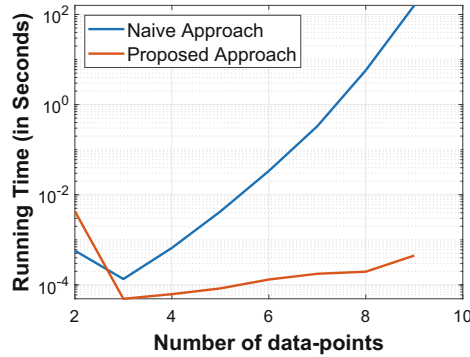


Fig. 5. The runtime of naive approach and the proposed approach to compute the sensitivity.

5 Conclusion and Future Work

In this paper, an approach based on the Hasse Diagram of an integer partition has been proposed, which requires $\mathcal{O}(n^3 p(n))$ time. There is an improvement in the time complexity compared to the naive approach, which requires us to compute B_n^2 *Precision* values. However, $p(n)$ still grows exponentially, so further improvement is needed. The sensitivity of *Precision* and *Recall* is bounded by $\mathcal{O}(n^4)$, so the improvement looks (not sure) realistic. It will be interesting to see whether the Hasse diagram can determine the sensitivity of other validity measures.

References

1. Aigner, M.: A characterization of the bell numbers. *Discret. Math.* **205**(1–3), 207–210 (1999)
2. Andrews, G.E., Eriksson, K.: *Integer Partitions*. Cambridge University Press (2004)
3. Antonir, A.C., Shapira, A.: Exact limit theorems for restricted integer partitions. *Adv. Math.* **407**, 108554 (2022)
4. Becker, H., Riordan, J.: The arithmetic of bell and stirling numbers. *Am. J. Math.* **70**, 385–394 (1948)
5. Campbell, J., Chern, S.: Nearly self-conjugate integer partitions. *New Zealand J. Math.* **54**, 1–7 (2023)
6. Canfield, E.R.: Integer partitions and the sperner property. *Theoret. Comput. Sci.* **307**(3), 515–529 (2003)
7. Christen, P., Hand, D.J., Kirielle, N.: A review of the f-measure: its history, properties, criticism, and alternatives. *ACM Comput. Surv.* **56**(3), 1–24 (2023)
8. Ghasemi, C., Yousefi, H., Shin, K.G., Zhang, B.: On the granularity of trie-based data structures for name lookups and updates. *IEEE/ACM Trans. Netw.* **27**(2), 777–789 (2019)
9. Hopkins, B., Sellers, J.A.: On blecher and knopfnacher’s fixed points for integer partitions. *Discret. Math.* **347**(5), 113938 (2024)

10. Kim, T., Kim, D.S.: A note on central bell numbers and polynomials. *Russ. J. Math. Phys.* **27**(1), 76–81 (2020). <https://doi.org/10.1134/S1061920820010070>
11. Li, W.: Approximation of the partition number after hardy and Ramanujan: an application of data fitting method in combinatorics. *arXiv preprint arXiv:1612.05526* (2016)
12. Mezo, I.: The r-Bell Numbers. *J. Integer Sequences* **14**(1), 1–14 (2011)
13. Mishra, S., Mondal, S., Saha, S.: Sensitivity - an important facet of cluster validation process for entity matching technique. In: Hameurlain, A., Küng, J., Wagner, R. (eds.) *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXIX*. LNCS, vol. 10120, pp. 1–39. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-54037-4_1
14. Robinson, J.P.: Edges in the poset of partitions of an integer. *J. Comb. Theory Ser. A* **48**(2), 236–238 (1988)
15. Shaffer, C.A.: *Data structures and algorithm analysis* (2022)
16. Wagner, S., Wagner, D.: *Comparing Clusterings: An Overview*. Universität Karlsruhe, Fakultät für Informatik Karlsruhe (2007)
17. Yeung, K.Y., Ruzzo, W.L.: Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics* **17**(9), 763–774 (2001)



Optimizing Software Release Management with GPT-Enabled Log Anomaly Detection

Jahanggir Hossain Setu, Md. Shazzad Hossain, Nabarun Halder,
Ashraful Islam^(✉), and M. Ashraful Amin

Center for Computational and Data Sciences, Independent University, Bangladesh,
Dhaka 1229, Bangladesh

{2312719, 2312720, ashraful, aminmdashraful}@iub.edu.bd,
shazzadshihab@gmail.com

Abstract. In the complex world of software systems, understanding and maintaining system stability and performance is of utmost significance. Finding anomalies in log data has become increasingly difficult due to these systems' growing complexity. Motivated by the need to improve software release management and ensure system reliability, this study exploits Generative Pretrained Transformer (GPT)-3's advanced word embedding and tokenizer functionalities to convert log data to adept at identifying atypical patterns and anomalies, delineated in a two-layered structure: offline and online layers. In the offline layer, historical log data undergoes processing through the GPT model, where it is divided into sentence and word embeddings. Sentence embeddings are clustered to generate labels and taggers for subsequent stages, while word embeddings directly create taggers for the online layer's sequence labeling. The online layer involves collecting real-time data, processing it through GPT to generate embeddings, and subjecting these embeddings to a sequence labeling process. This process yields templates and variables expediting the formation of train-test data splits for a classifier that detects anomalies. Different classifiers, namely Random Forest (RF), Light Gradient Boosting Machine (LightGBM), and Categorical Boosting (CatBoost), are evaluated. Experimental analysis on four distinct real-world datasets, namely Apache, BlueGene/L (BGL), Hadoop Distributed File System (HDFS), and Thunderbird, where CatBoost achieved remarkable accuracy rates of 99.75%, 99.00%, 98.75%, and 99.33%, respectively. The study also demonstrates that GPT-based embeddings provide a more effective anomaly detection solution than Bidirectional Encoder Representations from Transformers (BERT)-based embeddings. The proposed methodology is particularly designed to be integrated into software release management processes which enables automatic anomaly detection to augment quality control measures, thereby, expediting timely intervention.

Keywords: Log Data · Release Management · Anomaly Detection · Word Embeddings · Anomaly Detection · LLM · GPT-3 · Machine Learning

1 Introduction

Log anomaly detection is a crucial process in identifying and resolving issues in software systems. It involves analyzing log files generated by the components of the system to identify anomalies or patterns that may indicate errors, faults, or security threats [3]. In recent years, log anomaly detection has gained significant attention due to the increasing complexity of software systems and the growing number of security breaches [1]. Machine learning algorithms, including Support Vector Machine (SVM), Decision Tree, and Neural Networks, are popular for log anomaly detection due to their ability to learn from data and detect complex patterns [2]. Deep learning techniques, e.g., Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have also shown potential in this domain [4] along with Natural Language Processing (NLP) techniques, e.g., word embeddings and language models, such as, BERT and GPT-2 [6]. Challenges in log anomaly detection include high dimensionality, sparsity, and imbalanced data, which have been addressed with feature selection, dimensionality reduction, and data balancing techniques [7].

Log anomaly detection has various applications in software systems, including fault detection, security monitoring, and performance analysis. This study [8] is focused on the application of release management which includes release planning, configuration management, build and deployment, validation, distribution, and monitoring. Configuration management requires logging to track changes made to the software and ensure that all components of the release are configured correctly [9]. During build and deployment, logs are necessary to monitor progress and identify any errors or issues [10]. Logs are also critical during release validation to track testing results, including functional and non-functional testing, user acceptance testing, and system integration testing [10]. Finally, release monitoring requires log management to track the performance of the software after deployment and quickly resolve any anomalies or issues that may arise [11].

The followings are the major challenges associated with using release logs and log data anomaly detection techniques in the industry:

- One challenge is the sheer volume of log data generated during the release process [12]. Managing and analyzing this data can be a daunting task, especially for organizations that lack the necessary resources and expertise.
- Another challenge is the need to integrate different tools and systems used during the release process. For example, release management logs may be stored in one system, while build logs may be stored in another system. Integrating these logs and analyzing them together can be challenging [13].

Despite these challenges, many businesses are implementing release management and log data anomaly detection modules. For instance, Amazon employs log data anomaly detection techniques in its Amazon Web Services (AWS) infrastructure to detect security threats and anomalies [14]. In this study, a novel methodology for optimizing software release management using GPT-3 enabled

log anomaly detection is proposed. The proposed methodology accelerates release management by automatically spotting anomalies in log files, enabling early intervention to stop issues before they start.

The research evaluates the effectiveness of various classifiers, including Random Forest (RF), Light Gradient Boosting Machine (LightGBM), and Categorical Boosting (CatBoost), on four real-world datasets: Apache, BlueGene/L (BGL), Hadoop Distributed File System (HDFS), and Thunderbird. Remarkable accuracy rates achieved by CatBoost, reaching 99.75%, 99.00%, 98.75%, and 99.33%, respectively, underscore its efficacy in anomaly detection.

Furthermore, the methodology emphasizes the crucial role of K-means clustering in the offline layer of the proposed approach. Here, historical log data undergoes processing through the GPT model, generating both sentence and word embeddings. The sentence embeddings are then clustered using the K-means algorithm to generate labels and taggers for subsequent stages of the pipeline. This labeling process facilitates sequence labeling in the online layer, where real-time data is processed to detect anomalies. Additionally, the study demonstrates the superiority of GPT-based embeddings over Bidirectional Encoder Representations from Transformers (BERT) for anomaly detection. The proposed methodology offers an automated solution for detecting anomalies in log data, thereby enhancing software release management processes (Fig. 1).

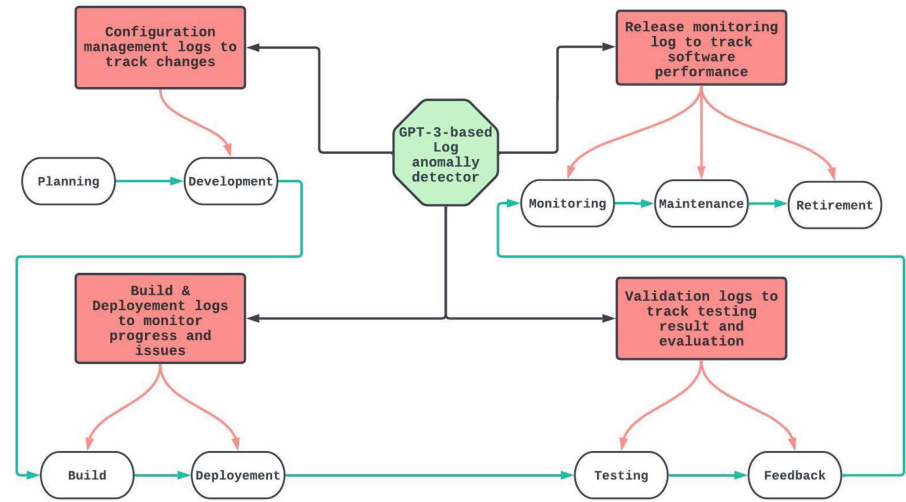


Fig. 1. GPT-based log anomaly detector in the release management pipeline

2 Literature Review

In this section, the state-of-the-art methods of detecting log anomalies are discussed and analyzed. These anomaly detection techniques are based on different

components, e.g., log parser, feature extraction, and classification models. The performance of these techniques is measured using different evaluation metrics, e.g., accuracy, precision, recall, and F1-measure. Additionally, the limitations of the log parsers from previous studies are presented in Table 1.

Table 1. Limitations of the log parsers from previous studies.

Log Parser	Research Gap
Drain [15]	<ul style="list-style-type: none">- No runtime analysis- Works well for fixed-dimension semantic vectors (Poor performance on Microsoft Service X)
Tokenization [16]	<ul style="list-style-type: none">- No runtime analysis- Log parser technique is not adaptive which led to biases and eventually poor performance for Blue Gene/L & Spirit dataget
Not Specified [17]	<ul style="list-style-type: none">- No runtime analysis- Log parser technique is not adaptive which led to poor performance for HDFS & BGL dataset
Log Mining [18]	<ul style="list-style-type: none">- The training of the model was focused on achieving optimal performance across all aspects- Authors considered simple architecture for TCN which led to a generalization problem
Log Sequence [19]	<ul style="list-style-type: none">- No runtime analysis- Very poor performance- Used single-sourced dataset- LQgUAD is suffering from a generalization problem

Zhang et al. [15] proposed a log anomaly detection method based on Drain, a popular log parser tool. They utilized Semantic Vectorization as a feature extractor and a Bidirectional-Long Short Term Memory (Bi-LSTM) classifier to detect anomalies in logs from HDFS and Microsoft Service X. Their results showed that the proposed method achieved a precision of 69%, recall of 99%, and an F1-measure of 81% for Microsoft Service X and a precision of 98%, recall of 100%, and an F1-measure of 99% for HDFS. However, the proposed method lacked a runtime analysis, and it performed poorly on Microsoft Service X due to the fixed-dimension semantic vector. This research gap highlights the need to improve the scalability of LAD methods.

Nedelkoski et al. [16] put forward a log anomaly detection framework based on tokenization as a log parser, Multi-Head Self-Attention Transformer as a feature extractor, and classification of logs from BGL, Thunderbird, and Spirit datasets. Their results showed that the proposed method achieved high accuracy, precision, recall, and F1-measure for the Thunderbird dataset, with 100% accuracy, 99% precision, 100% recall, and 99% F1-measure, and for Blue Gene/L, with 95% accuracy, 52% precision, 87% recall, and 65% F1-measure.

Guo et al. [17] proposed LogBERT; a LAD method based on BERT; as a feature extractor and classifier of logs from HDFS, BGL, and Thunderbird

datasets. Their results showed that the proposed method achieved high precision, recall, and F1-measure for all three datasets, with the highest F1-measure of 0.9664 for the Thunderbird dataset. Both [16,17] found that their proposed LAD methods achieved lower performance for certain datasets due to the non-adaptive log parser technique used. Nedelkoski et al. observed lower performance for the Spirit dataset, while [17] observed lower F1-measure for HDFS and BGL datasets. Furthermore, both studies identified a research gap related to the lack of runtime analysis, highlighting the need for future research in this area.

Wang et al. [18] developed LightLog, a log anomaly detection technique that utilizes log mining as a log parser, Word2Vec and a Post Processing Algorithm (PPA) for feature extraction, and a Temporal Convolutional Network (TCN) for classification. One advantage of this technique is its high precision and recall rates, achieving an F1-measure of 97% for both HDFS and BGL datasets. Another advantage is its relatively fast detection time, taking only 4.3 min for 100 samples in the HDFS dataset and 0.45 min for 100 samples in the BGL dataset. However, the authors also noted that the model was trained to achieve the best overall performance and the simple architecture used for TCN may result in a generalization problem, which could impact its performance on other datasets.

Wang et al. [19] presented another log anomaly detection approach called Log Unsupervised Anomaly Detection (LogUAD), which uses log sequence data and the Word2Vec feature extraction method followed by the K-Means clustering algorithm. However, the study had several limitations, including the lack of runtime analysis and very poor performance with an F1-measure of only 67.25% on the BGL dataset. Additionally, the authors only used a single-source dataset, limiting the generalizability of the LogUAD approach.

This highlights the research gap in log anomaly detection where more advanced techniques need to be developed to handle the large-scale and diverse log data generated by modern computer systems.

3 Materials and Methodology

The proposed methodology in Fig. 2 consists of two layers: offline and online. In the offline layer, historical log data is given to the GPT model, which separates the data into sentence and word embeddings. The sentence embeddings pass through a clustering process based on K-Means algorithm which generates labels and create taggers for the next layer. Word embeddings, on the other hand, create taggers directly and these taggers are connected to the sequence labeling stage of the online layer. The taggers trained on clustered data assist in labeling new data points during the sequence labeling stage, where patterns and structures are identified and assigned labels. This labeling process is pivotal as it enables subsequent classification of anomalies based on the identified patterns.

In the online layer, real-time data is collected from the database, and processed by GPT, which generates embeddings for the data. The embeddings pass through the sequence labeling process that produces templates and variables, used to make train-test data splits fed into the classifier to detect anomalies.

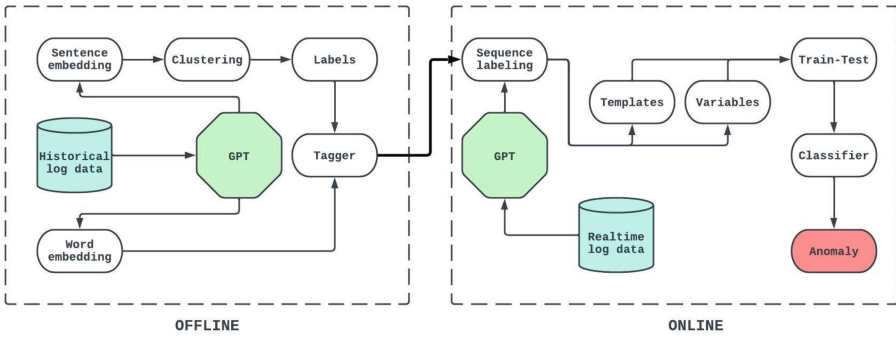


Fig. 2. Proposed GPT-based log anomaly detection framework

This approach combines machine learning methods, such as classification and clustering with NLP methods, e.g., word and sentence embeddings and sequence labeling to find anomalies in log data. To label new data points, taggers are trained on the clustered data. Sequence labeling uses patterns and structures to assign labels to each element in a sequence of data, whereas classifiers are used to predict the class or label of new data points. Utilizing the strength of GPT to generate embeddings with clustering, sequence labeling, and classification techniques for anomaly detection, this methodology offers a comprehensive solution for log anomaly detection.

3.1 Dataset

The log datasets used in this study encompass a diverse range of environments and applications. Each dataset offers unique insights into system behavior and operational patterns. The Apache dataset consists of logs generated by the Apache web server. It contains client requests, server responses, errors, and other operational metrics essential for web server management and analysis [25]. The BGL dataset originates from the Blue Gene/L supercomputer system. It contains system events, errors, and performance metrics critical for understanding and optimizing high-performance computing environments [25]. The HDFS dataset includes records of system events, file operations, and cluster interactions within distributed computing clusters [25]. The Thunderbird Logs Dataset consists of logs from the Thunderbird email client, documenting user interactions, email processing events, errors, and application performance metrics. All these datasets are publicly available [25]. Table 2 presents detailed information of the four datasets used in this study. Each dataset varies in size, attributes, and data source, all of which have specific implications for their application in data analysis and machine learning tasks.

Table 2. Finding gaps from previous studies

Dataset Name	Instance Size	Total data attributes	Data source	Implication
Apache	2000	6	Web server access logs	Anomaly detection, classification
BGL	2000	13	Supercomputer system log	Anomaly detection, log parsing
HDFS	2000	14	Hadoop Distributed File System logs	Anomaly detection, log analysis
Thunderbird	2000	14	Email client application logs	Log parsing, anomaly detection

3.2 Generating Word and Sentence Embedding from GPT-3

This approach employs a technique that splits the textual information into smaller portions or batches. Each batch is processed independently for computational efficiency. Embedding generation is applied to each batch of data which utilizes the GPT-‘ENGINE’ [5] to transform the textual content of the dataset into embeddings. These embeddings are vector representations of words and sentences, capturing the semantic meaning and context of the text. Two types of embeddings are generated in parallel - word embeddings and sentence embeddings. Word embeddings represent individual words within the text and capture their semantic relationships with other words. On the other hand, sentence embeddings represent the overall meaning or context of entire sentences. After processing all batches, the generated word embeddings and sentence embeddings are concatenated into two separate arrays. This consolidation allows for a unified representation of the entire dataset. Time taken to complete this process depends on the size of the dataset and the efficiency of the embedding method. By splitting the data into manageable chunks and processing them in parallel, it helps streamline the embedding [20] generation process, making it suitable for various NLP applications. The time required for completion depends on dataset size and processing efficiency.

3.3 Clustering with K-Means and Labeling

The application of the K-Means clustering algorithm to a set of data points is represented as sentence embedding [21]. The K-Means algorithm is initialized with a predefined number of clusters, which, in this case, is set to two. This means that the algorithm will aim to partition the data into two distinct groups or clusters. In the data fitting step, the algorithm proceeds to fit the sentence embedding data to the K-Means model. During this step, K-Means attempts to find two centroids in the feature space that will act as representative points for each of the two clusters. These centroids are initially placed randomly. After fitting the data, K-Means iteratively refines the positions of the centroids to minimize the sum of squared distances between data points and their assigned centroids. This process involves assigning each data point to the nearest centroid,

which effectively divides the data into two clusters. The algorithm repeats this assignment and centroid adjustment until convergence, aiming to find centroids that minimize the intra-cluster distance. By the end of the K-Means process, each data point has been assigned to one of the two clusters. The assignment is based on the nearest centroid. In this way, K-Means effectively labels each data point as belonging to one of the two identified clusters. The centroids are obtained using the K-means cluster centres attribute [22].

3.4 Tagging and Sequence Labeling

Parts Of Speech (POS) tagging is a NLP technique that assigns grammatical labels or tags to individual words within a text. These tags represent the part of speech of each word, such as nouns, verbs, adjectives, etc. [23]. This technique utilizes a predefined model to assign POS tags to each word in the dataset. The result is a list of word-POS tag pairs, where each word in the embedding dataset is associated with its corresponding part-of-speech tag. Sequence labeling is a broader technique that involves assigning labels to elements within a sequence of data. In this case, it appears that we are performing sequence labelling on a combination of word embeddings and sentence embeddings where the sequence label variable is created by concatenating the word and sentence embeddings. This operation combines the embeddings of individual words and entire sentences into a single sequence. Subsequently, it seems that the sequence label is concatenated with itself, which might be intended to create a larger sequence for sequence labelling [24].

3.5 Training and Evaluation

The test size was set to 0.2 which means that 20% of the data will be allocated to the test set leaving 80% for the training set. This division ensures that a portion of the data is reserved for evaluating the model's performance on unseen data. Three different classifiers, e.g., RF, LightGBM, and CatBoost, were utilized and these classifiers were trained on a subset of the data and then evaluated on data it has never seen before to determine its generalization ability. The evaluation metrics used for analysis are recall (macro average), precision (macro average), accuracy, F1-measure (macro average), Area Under Curve (AUC) score, and classification time (seconds). Recall measures the proportion of actual positive samples that are correctly classified as positive by the model. Precision, on the other hand, measures the proportion of positive predictions that are actually positive. Accuracy is the proportion of total correct predictions made by the model. F1-measure is the harmonic mean of Recall and Precision. AUC score measures the performance of the model in distinguishing between positive and negative samples, while Classification Time (s) measures the time taken by the model to classify the data.

3.6 Experimental Setup

For the experimental setup, a specific workstation configuration and a range of software packages and libraries were utilized to ensure optimal performance and compatibility. The detailed specifications of the hardware and software used in the experiments are summarized in Table 3.

Table 3. Summary of workstation configuration and software packages used in the experimental setup

Workstation Cofiguration	OS	MacOS X El Captain 10.11.6
	CPU	Intel Core i5 @1.6 GHz
	Memory	4 GB 1600 MHz DDR3
	SecondaryStorage	SSD 128GB
	GPU	Intel HD Graphics 6000 1536 MB
Software packages and libraries	Google Colab	
	Python 3.10	
	KERAS 2.4.3	
	TensorFlow 2.3.0	
	Hugging Face Transformers	

3.7 Threads to Validity

The validity of the experimental findings is subject to several potential limitations. Firstly, the utilization of publicly available datasets may not fully encapsulate the intricacies and scale of real-world data scenarios. Consequently, the diversity and richness of data patterns and anomalies present in actual environments may not be fully represented. Moreover, the datasets employed in the experiment might lack diversity, potentially overlooking certain data patterns and anomalies prevalent in real-life situations. This limitation could impact the generalizability of the results across different domains and industries.

Additionally, variability in data quality and characteristics across various real-world datasets may introduce biases and confounding factors that were not fully accounted for in the experimental setup. Consequently, the generalizability of the findings to broader contexts may be limited. Furthermore, the experimental pipeline’s performance may vary when applied to different domains or industries, underscoring the necessity for further investigation and adaptation to specific use cases.

Moreover, external factors such as technological advancements and shifts in data distribution over time could influence the effectiveness and applicability of the experimental findings in real-world scenarios. Lastly, despite efforts to design a robust experimental setup and methodology, certain nuances or intricacies specific to particular datasets or use cases may have been overlooked, potentially constraining insights into the pipeline’s capabilities in real-world settings.

4 Result Analysis and Discussion

Table 4 provides the performance evaluation of GPT. The evaluation of the classifiers on both the Apache and BGL datasets revealed the high accuracy of all three classifiers, as well as the superior performance of the CatBoost classifier in terms of recall, precision, accuracy, F1-measure, and AUC score. Despite the longer classification time of the CatBoost classifier compared to LightGBM and RF, the latter demonstrated superior performance in achieving the fastest classification time while maintaining good performance in all evaluation metrics, including a high AUC score. When evaluating the classifiers on the HDFS and Thunderbird datasets, it was found that their performance was exceptional as evidenced by their high accuracy, recall, precision, F1-measure, and AUC scores. RF stood out as the best performer in terms of recall and precision on both datasets. However, CatBoost demonstrated the fastest classification time on both datasets, while all three classifiers maintained their excellent performance across all evaluation metrics.

Table 4. Performance evaluation based on GPT embeddings

Dataset	Classifier	Recall (%)	Precision (%)	Accuracy (%)	F1- Measure (%)	AUC Score (%)	Classification Time (s)
Apache	RF	97.94	98.03	98.00	97.98	99.50	0.024524
	LightGBM	98.47	98.52	98.50	98.49	98.80	0.007038
	CatBoost	99.74	99.76	99.75	99.75	99.80	0.010937
BGL	RF	98.77	98.73	98.75	98.75	99.30	0.019798
	LightGBM	97.76	97.74	97.75	97.75	99.50	0.025235
	CatBoost	98.99	98.99	99.00	98.99	99.90	0.005348
HDFS	RF	98.72	98.78	98.75	98.75	98.90	0.027765
	LightGBM	96.79	97.21	97.00	96.96	98.90	0.035475
	CatBoost	97.38	97.52	98.75	97.45	98.40	0.007264
Thunderbird	RF	99.31	99.35	99.33	99.33	99.40	0.024687
	LightGBM	98.97	99.02	99.00	98.99	99.40	0.022166
	CatBoost	99.13	99.19	99.33	99.16	99.60	0.005213

The findings demonstrate that CatBoost outperforms the other two classifiers, exhibiting exceptional performance in terms of recall, precision, accuracy, F1-measure, and AUC score. Despite a negligible difference in classification time compared to LightGBM and RF (Apache dataset), the superiority of CatBoost in identifying both positive and negative instances correctly and with high precision, combined with its overall excellent performance of testing presented in Fig. 3 and overall model performance in Fig. 4, makes it the most favourable choice. These results indicate that CatBoost is a highly efficient and effective classifier that can be deployed for log anomaly detection.

On the other hand, Table 5 shows the performance of the classifiers trained on the same datasets but using BERT-based embeddings. It can be seen that the performance of the classifiers is generally lower compared to the results obtained using GPT-based embeddings. CatBoost still performs the best in most of the datasets, with LightGBM following closely behind. RF, however, performs

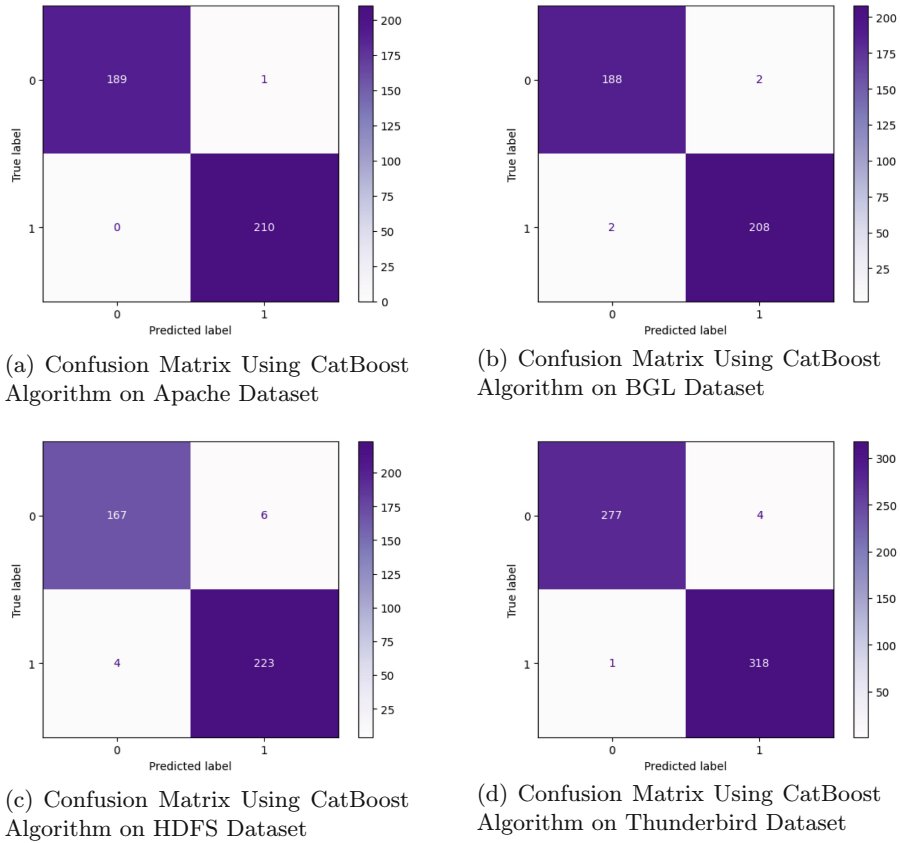
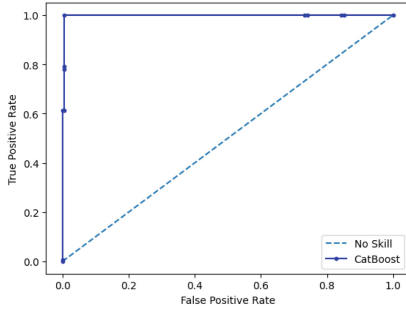


Fig. 3. Confusion matrices for CatBoost algorithm on different datasets

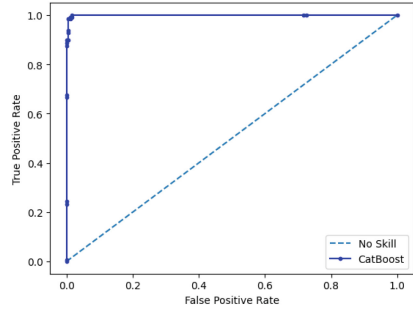
Table 5. Performance evaluation based on BERT embeddings

Dataset	Classifier	Recall (%)	Precision (%)	Accuracy (%)	F1- Measure (%)	AUC Score (%)	Classification Time (s)
Apache	RF	93.76	92.06	93.39	92.91	92.12	0.076543
	LightGBM	94.81	92.28	94.35	93.545	92.19	0.021876
	CatBoost	95.26	94.56	95.44	94.91	92.91	0.019702
BGL	RF	94.89	94.52	94.80	94.705	94.35	0.098656
	LightGBM	94.11	94.76	94.98	94.435	94.63	0.065540
	CatBoost	95.00	94.98	95.33	94.99	94.99	0.012765
HDFS	RF	93.45	94.65	94.44	94.05	92.26	0.045776
	LightGBM	95.88	94.97	93.21	95.425	95.62	0.039908
	CatBoost	95.78	96.45	95.55	96.115	95.97	0.009876
Thunderbird	RF	95.44	94.77	95.66	95.105	95.98	0.067750
	LightGBM	95.90	94.56	95.11	95.23	95.23	0.053012
	CatBoost	95.87	96.23	96.75	96.05	99.69	0.010087

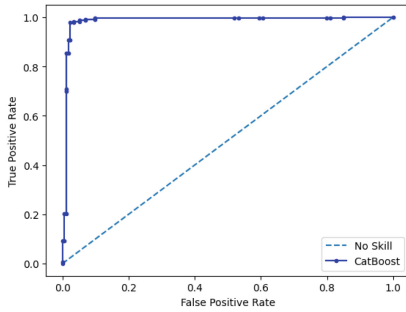
poorly in all the datasets. The classification time is relatively higher compared to Table 4, with RF taking the longest time.



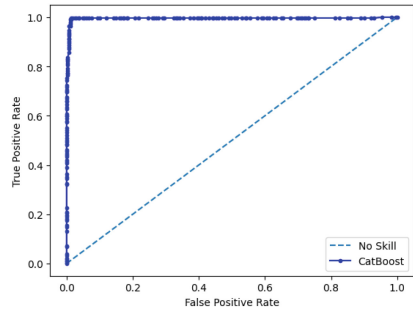
(a) ROC Curve Using CatBoost Algorithm on Apache Dataset



(b) ROC Curve Using CatBoost Algorithm on BGL Dataset



(c) ROC Curve Using CatBoost Algorithm on HDFS Dataset



(d) ROC Curve Using CatBoost Algorithm on Thunderbird Dataset

Fig. 4. ROC curves for CatBoost algorithm on different datasets

The results discussed above indicate that GPT-based embeddings are more effective than BERT-based embeddings for the datasets used in the study. This can be attributed to the fact that GPT is a more recent and advanced language model architecture than BERT. Furthermore, the performance of the classifiers varied significantly across the different datasets. For instance, in Apache dataset, GPT embeddings improved CatBoost’s accuracy by 4.31% compared to BERT, whereas in BGL dataset, the improvement was 2.77%. These variations suggest that the characteristics of the datasets play an important role in determining the effectiveness of the embeddings and classifiers used.

5 Conclusion and Future Works

In this study, a comprehensive evaluation of various classifiers using both GPT-based and BERT-based embeddings for log anomaly detection in the software release pipeline was conducted across multiple datasets. The analysis yielded significant findings, including the superior performance of the CatBoost classifier in terms of recall, precision, accuracy, F1-measure, and AUC score when using

GPT-based embeddings. This consistent excellence across datasets underscores CatBoost as the preferred choice for log anomaly detection. Furthermore, the study highlighted the influence of pre-trained language model architectures on the effectiveness of embeddings. GPT-based embeddings generally outperformed their BERT-based counterparts, emphasizing the importance of the choice of language model. Additionally, the variability in classifier performance across datasets emphasized the significance of dataset characteristics in shaping classifier and embedding choices. Each dataset's unique patterns and anomalies can impact model performance, and such considerations are crucial when designing log anomaly detection systems. However, it is essential to note that while CatBoost demonstrated superior performance, it also incurred slightly longer classification times in certain scenarios. This prompts practitioners to make a thoughtful trade-off between classification speed and performance, depending on specific use cases and resource constraints.

It is essential to acknowledge a potential challenge that may arise in the future, particularly in the context of software release management. As software release management processes evolve, the volume of log data generated during each batch of releases is expected to increase significantly. This expansion can lead to longer processing times during the embedding stage, as new embeddings must be generated for the additional data. Also, to address the complexity and scalability of advanced machine learning and NLP techniques, we propose several strategies. These include using pre-trained models and cloud-based services to simplify deployment and reduce technical expertise requirements. We emphasize modular architecture and distributed computing for scalability, with examples, e.g., AWS and Google Cloud for managing data and dynamic changes. Our approach can be integrated into existing software release management systems through Application Programming Interfaces (APIs) or middleware etc. This integration allows for seamless monitoring and anomaly detection during the software release cycle which provides real-time feedback and prompt responses to potential issues. Future research may focus on optimizing the efficiency of embedding generation processes or exploring strategies for handling large volumes of log data without compromising the overall performance of log anomaly detection systems. This adaptive approach will be crucial in ensuring that log anomaly detection remains a valuable tool in the ever-growing landscape of software release management. Future work will also certainly benefit from broader access to diverse log datasets to further validate and extend the findings of our study.

References

1. Le, V.-H., Zhang, H.: Log-based anomaly detection with deep learning: how far are we? In: Proceedings of the 44th International Conference on Software Engineering, pp. 1356–1367. IEEE (2022)
2. Naseer, S., et al.: Enhanced network anomaly detection based on deep neural networks. *IEEE Access* **6**, 48231–48246 (2018)

3. Cao, Q., Qiao, Y., Lyu, Z.: Machine learning to detect anomalies in web log analysis. In: 2017 3rd IEEE international conference on computer and communications (ICCC), pp. 519–523. IEEE (2017)
4. Tziolas, T., Papageorgiou, K., Theodosiou, T., Papageorgiou, E., Mastos, T., Papadopoulos, A.: Autoencoders for anomaly detection in an industrial multivariate time series dataset. *Eng. Proc.* **18**(1), 23 (2022)
5. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
6. Zhao, Z., Niu, W., Zhang, X., Zhang, R., Yu, Z., Huang, C.: Trine: syslog anomaly detection with three transformer encoders in one generative adversarial network. *Appl. Intell.*, 1–10 (2021). <https://doi.org/10.1007/s10489-021-02863-9>
7. Huang, S., Liu, Y., Fung, C., Wang, H., Yang, H., Luan, Z.: Improving log-based anomaly detection by pre-training hierarchical transformers. *IEEE Trans. Comput.* **72**, 2656–2667 (2023)
8. Santosa, I., Mulyana, R.: The IT services management architecture design for large and medium-sized companies based on ITIL 4 and TOGAF framework. *JOIV: Int. J. Inform. Vis.* **7**(1), 30–36 (2023)
9. Yu, G., et al.: LogReducer: identify and reduce log hotspots in kernel on the fly. In: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), pp. 1763–1775. IEEE (2023)
10. Abbas, M., Hamayouni, A., Moghadam, M.H., Saadatmand, M., Strandberg, P.E.: Making Sense of Failure Logs in an Industrial DevOps Environment. In: Latifi, S. (eds.) *International Conference on Information Technology-New Generations*, pp. 217–226. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-28332-1_25
11. Kauffman, S.: Log analysis and system monitoring with NFER. *Sci. Comput. Program.* **225**, 102909 (2023)
12. Zanella, R., Welch, B., Mendelsohn, M., Korte, B.: Enterprise Log Managers: An Unsexy. But Vital, Tool-Global Security Mag Online (2023)
13. Meng, W., et al.: LogSummary: unstructured log summarization for software systems. *IEEE Trans. Netw. Serv. Manag.* **20**, 3803–3815 (2023)
14. Li, M., Sun, M., Li, G., Han, D., Zhou, M.: MDFULog: multi-feature deep fusion of unstable log anomaly detection model. *Appl. Sci.* **13**(4), 2237 (2023)
15. Zhang, X., et al.: Robust log-based anomaly detection on unstable log data. In: *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 807–817 (2019)
16. Nedelkoski, S., Bogatinovski, J., Acker, A., Cardoso, J., Kao, O.: Self-attentive classification-based anomaly detection in unstructured logs. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 1196–1201. IEEE (2020)
17. Guo, H., Yuan, S., Wu, X.: LogBERT: log anomaly detection via BERT. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2021)
18. Wang, Z., Tian, J., Fang, H., Chen, L., Qin, J.: LightLog: a lightweight temporal convolutional network for log anomaly detection on the edge. *Comput. Netw.* **203**, 108616 (2022)
19. Wang, J., Zhao, C., He, S., Gu, Y., Alfarraj, O., Abugabah, A.: LogUAD: log unsupervised anomaly detection based on Word2Vec. *Comput. Syst. Sci. Eng.* **41**(3), 1207 (2022)
20. Borders, T.L., Volkova, S.: An introduction to word embeddings and language models (No. INL/EXT-21-61935-Rev000). Idaho National Lab.(INL), Idaho Falls, ID, United States (2021)

21. Kusumaningrum, R., Khoerunnisa, S.F., Khadijah, K., Syafrudin, M.: Exploring community awareness of mangrove ecosystem preservation through sentence-BERT and K-Means clustering. *Information* **15**(3), 165 (2024)
22. Fan, J., Huang, L., Gong, C., You, Y., Gan, M., Wang, Z.: KMT-PLL: K-means cross-attention transformer for partial label learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 1–2 (2024)
23. Tehseen, A., Ehsan, T., Liaqat, H.B., Ali, A., Al-Fuqaha, A.: Neural POS tagging of shahmukhi by using contextualized word representations. *J. King Saud Univ. Comput. Inf. Sci.* **35**(1), 335–356 (2023)
24. Zhang, C., et al.: LayerLog: log sequence anomaly detection based on hierarchical semantics. *Appl. Soft Comput.* **132**, 109860 (2023)
25. Zhu, J., He, S., He, P., Liu, J., Lyu, M.R.: Loghub: a large collection of system log datasets for AI-driven log analytics. In 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), pp. 355–366. IEEE (2023)



Patch-Based Prototypical Cross-Scale Attention Network for Anomaly Detection

Tung-Lin Wang, Jun-Wei Hsieh^(✉), and Yi-Kuan Hsieh

College of AI and Green Energy, National Yang Ming Chiao Tung University,
Tainan, Taiwan

jwhsieh@nycu.edu.tw

Abstract. Anomaly detection and localization play crucial roles in industrial manufacturing to help maintain product quality and minimize defects. However, anomalies are rare and challenging to collect, leading to imbalance data that cause a biased model to be trained and sensitive to noisy or irrelevant features. In addition, anomalies are often subtle, diverse, and change over time, making them difficult to differentiate, further complicating the detection and localization tasks. To address these challenges, we propose a new Patch-based Prototypical Cross-Scale Attention Network (PPCA-Net) to effectively identify anomaly regions by learning residual features across different scales and sizes, distinguishing abnormal from normal patterns. It consists of two key components: the Scale-Aware Channel Attention Module (SACAM) and the Patch-based Cross-Scale Attention Module (PCSAM). These modules facilitate interactive feature inferences across multiple scales, significantly enhancing the ability to capture abnormal features of various sizes in various environments. Furthermore, we incorporate diverse anomaly generation strategies, including multi-scale prototypes to better represent feature disparities between abnormal and normal patterns, thereby enhancing overall effectiveness. Through extensive experimentation on the challenging MVTec AD [1] benchmark, PPCA-Net demonstrates superior performance in both unsupervised and supervised methods, highlighting its effectiveness in anomaly identification.

Keywords: Anomaly Detection · Anomaly segmentation · defect inspection

1 Introduction

Industrial defects have the potential to wreak havoc on production processes, leading to compromised product quality, decreased productivity, and even equipment damage or stoppages. As a result, there is a heightened emphasis on leveraging anomaly detection (binary classification of images) and localization (binary classification of pixels) technologies to promptly identify and rectify these issues, ultimately aiming to enhance both product quality and manufacturing efficiency. They can not only aid in defect detection within industrial environments, but

also show remarkable progress in areas such as medical image analysis and video surveillance.

Given the profound impact of industrial defects on product quality and production efficiency, considerable research efforts have been dedicated to addressing anomaly detection and anomaly localization. However, effectively tackling both of these challenges simultaneously remains a formidable task for many researchers in the field. A primary hurdle encountered by researchers is the excessively low scrap rate prevalent in industrial settings. This low occurrence of defective samples poses a significant challenge when it comes to utilizing them to train deep neural networks. In contrast, non-defective samples are often abundant in practical industrial production environments, significantly outnumbering their defective counterparts. This stark contrast in sample availability further complicates the training process for anomaly detection models.

This challenge is compounded by the labor-intensive and costly process of labeling defect samples. In supervised learning approaches, where accurate annotations are vital for model training, detection networks require precise bounding box labels, while classification networks rely on comprehensive image-level annotations. Achieving high-quality labeling is crucial, as the performance of defect detection models is heavily dependent on the quality of annotated data. Consequently, specialized professionals are often needed to meticulously annotate defects, especially in specialized industrial domains with unique defect characteristics. In real-world industrial settings, defects exhibit a wide range of types and characteristics that are often unpredictable. This unpredictability makes pre-labeling of defects impractical, as it's impossible to anticipate every possible manifestation of defects. Consequently, traditional approaches relying on pre-labeled datasets may not fully capture the diverse spectrum of real-world defect scenarios. Even the random generation of anomalies, intended to diversify datasets, may not consistently improve overall model performance. The inherent variability and complexity of real-world defects cannot always be accurately replicated through random generation techniques. Therefore, there's a critical need to develop robust detection systems capable of accurately identifying surface defects, especially in situations where diverse defect samples are limited.

Recently, several supervised methods have been introduced. DeepSAD [17] adjusts the boundary between anomalies and the single-class center in latent space to restrict known anomalies and obtain more compact single-class descriptors. On the other hand, DRA [5] and DevNet [13] formulate anomaly detection as a Multiple Instance Learning (MIL) problem, considering an image as anomalous if any region contains a defect. Through MIL learning at the patch level, these methods effectively reduce interference from normal patches in images. However, supervised methods may face challenges when dealing with anomalies that occupy only a small portion of the image. Due to insufficient availability of anomaly samples for training, unsupervised models have gained prominence in current research. These models primarily address data imbalance by learning the feature distribution of normal samples or generating synthetic anomaly samples. However, they often exhibit poor performance when confronted with real-world anomalies, leading to a high rate of false positives and false negatives, thus com-

promising recognition accuracy. Furthermore, these erroneous predictions can result in costly false alarms or missed detections in industrial and manufacturing settings, where even minor anomalies can significantly affect product quality and production efficiency.

This paper presents a novel framework, named Patch-Based Prototypical Cross-Scale Attention Network (PPCA-Net) designed to tackle the challenges inherent in anomaly detection and localization. Firstly, we introduce multi-scale prototypes into PPCA-Net to represent normal patterns. Unlike previous methods, which often construct normal patterns from concatenated feature maps or randomly sampled feature maps, PPCA-Net constructs these patterns using prototypes derived from intermediate feature maps of different scales. This approach preserves spatial details and offers more precise and representative normal patterns. Furthermore, we compute feature map residuals by assessing the deviation between anomalous images and the closest prototype at each scale. We also integrate multi-scale fusion blocks to facilitate information exchange across different scales. Recognizing the significant variability in the appearance of anomaly regions, two new modules are introduced in this paper: the Scale-Aware Channel Attention Module (SACAM) and the Patch-based Cross-Scale Attention Module (PCSAM). These modules enable effective detection of patch-level inconsistencies between feature maps across various scales. In contrast to previous methods relying on image-level supervision, our model learns to segment anomaly maps with pixel-level supervision, emphasizing anomalous regions and enhancing generalization. In addition, we propose several anomaly generation strategies to address data imbalance issues and improve anomaly diversity. With these innovations, our method achieves more accurate localization compared to previous unsupervised and supervised approaches.

The main contributions of this paper are summarized as follows:

- We propose a new Patch-based Prototypical Cross-Scale Attention Network (PPCA-Net) for effective anomaly detection and localization by learning residual features across different scales and sizes.
- SACAM supports cross-scale channel weight calculation, enabling global highlighting of significant channels.
- PCSAM divides the feature map into patches with narrower fields of view, facilitating the extraction of smaller and finer-grained features. This attention mechanism helps to better differentiate between anomalous and normal patches, thereby enhancing performance.
- Extensive experiments conducted on the MVTec AD dataset demonstrate the excellent performance of our method in anomaly detection.

2 Relative Work

2.1 Supervised Approaches

Supervised anomaly detection hinges on leveraging known anomaly data to refine the relationship between anomalies and normal samples, thus increasing detection accuracy and efficiency. Unlike traditional unsupervised methods,

supervised approaches benefit from labeled anomaly samples during training, enabling models to better discern and generalize. Some studies have employed single-class classification techniques [18], focusing on learning from a limited set of anomaly instances. For example, methodologies such as DevNet [13] utilize the distribution of normal samples to recalibrate samples containing anomalies, thereby enhancing anomaly detection capabilities. Furthermore, certain approaches employ multi-head models [5] to decipher complex anomaly representations, with each head specializing in specific anomaly types, thus enhancing model diversity and generalization performance. However, supervised methods face challenges and limitations. Imbalanced data distributions can lead to overfitting on a small subset of anomaly instances, reducing performance in detecting unseen anomalies. Additionally, when dealing with subtle anomalies, models struggle with the abundance of normal sample information, hindering their ability to accurately identify anomalies of varying sizes and shapes, thereby diminishing detection accuracy and reliability. Therefore, advancing supervised methods requires addressing these challenges through improved sample balancing strategies and more representative feature representations, with the aim of strengthening the resilience and generalization of the model. Continuous technological progress and deeper research efforts are expected to elevate supervised methods to a more prominent role in the anomaly detection domain, yielding more satisfactory results.

2.2 Unsupervised Approaches

Unsupervised methods rely exclusively on normal data during training. For example, autoencoder-based techniques [3] continuously train on normal samples to reconstruct them, aiming to detect anomalies by identifying discrepancies between input and reconstructed images, thus highlighting anomalous regions. Despite efforts to enhance performance by introducing generative models such as Variational Autoencoders (VAE) [10] and Generative Adversarial Networks (GAN) [8], they often struggle to accurately generate normal regions in images, resulting in rough reconstructions and false anomaly detections. Other approaches, like those based on normalizing flows [16], incur significant computational costs.

Knowledge distillation-based methods [2] compare features between teacher and student networks to detect anomalies, while deep feature modeling methods identify anomalies by comparing features extracted from images. Self-supervised learning methods [14] devise proxy tasks to predict or recover hidden regions or attributes in images. One-class classification methods [18] aim to map training data to a small hypersphere in the feature space, making anomaly samples more challenging to identify. However, they demonstrate suboptimal performance when confronted with subtle and challenging anomalies.

3 Method

We present a novel framework named Patch-based Prototypical Cross-Scale Attention Network (PPCA-Net) to tackle the challenges inherent in anomaly

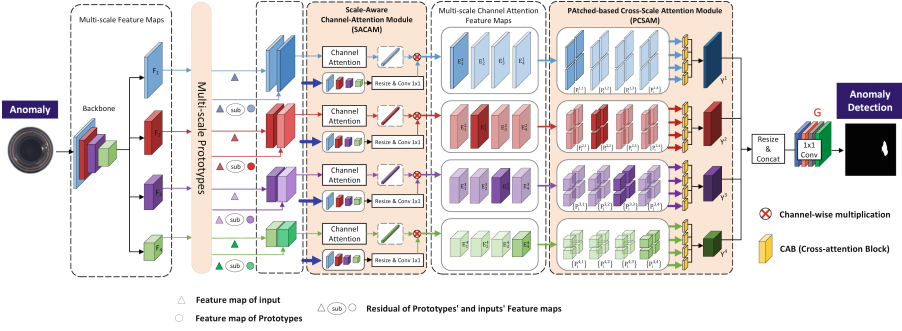


Fig. 1. The proposed Patch-based Prototypical Cross-Scale Attention Network (PPCA-Net) consists of two main modules. SACAM strengthens the global channel importance by examining multi-scale feature maps. PCSAM improves the detection and segmentation of small anomalous regions by leveraging cross-scale attention among patches.

detection and localization. In order to overcome the limitations of current supervised anomaly detection methods, we introduce two innovative approaches for handling multi-scale features, resulting in the creation of two new modules: (1) The Scale-Aware Channel Attention Module (SACAM) focuses on computing scale-aware attention within feature channels, enabling the effective detection of anomalies across a range of sizes. (2) The Patch-based Cross-Scale Attention Module (PCSAM) calculates attention from deep features among patches, facilitating improved detection accuracy for small anomalies. The integration of these two modules forms the Patch-based Cross-Scale Attention Network (PPCA-Net), which combines the SACAM and PCSAM modules into a unified pipeline and can significantly improve the accuracy of anomaly detection and localization. The main architecture, depicted in Fig. 1, utilizes ResNet-18 [7] as the encoder and incorporates multiscale prototypes, in addition to our proposed SACAM and PCSAM modules for anomaly classification. Subsequently, we will delve into the specifics of the design.

3.1 Multiscale Prototypes

Given a set of training samples $\{X_i\}$ for anomaly detection, only two classes are used to classify them, *i.e.*, normal or abnormal. However, there are various normal and abnormal types. To obtain better classification results, multiple normal and abnormal prototypes must be constructed to classify $\{X_i\}$. Given a CNN backbone, we convert each training sample X_i to a set of feature maps $\{F_n^i\}$, where F_n^i denotes the feature map of X_i on the n th scale. For a class C_y ($y = 0$ for normal and $y = 1$ for abnormal), we use the ISO-data method [11] to classify all training samples in C_y into different prototypes $\{\mathbf{P}_{j,n}^y\}$, where $\mathbf{P}_{j,n}^y$ denotes the j th prototype at the n th scale for C_y . Due to the nature of the ISO-data method, we do not need to consider setting the number of prototypes to classify the samples in C_y . Furthermore, we employ four scales of prototypes

Channel Attention Block (CHAB)

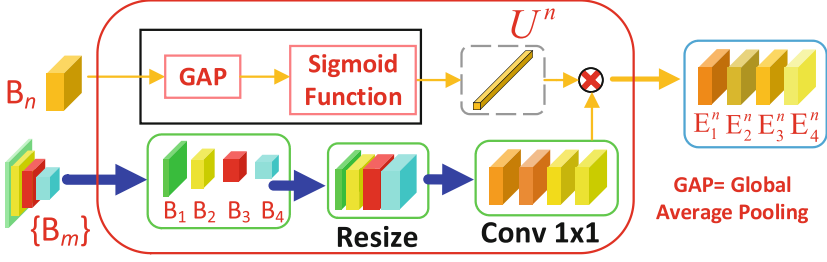


Fig. 2. The CHannel Attention Block (CHAB). U_n is obtained by a GAP (Global Average Pooling) operation followed by a Sigmoid function. All $\{B_m\}$ are resized and convoluted to have the same size and number of channels as B_n . For each resized B_n , their channels are weighted by U_n to form a new feature map E_m^n .

($n \in \{1, 2, 3, 4\}$). After clustering, each scale's prototype $\mathbf{P}_{j,n}^y$ is fixed during the subsequent model training process. Given a training sample $X_i \in C_y$ and its feature map F_n^i at the n th scale, we can find the closest prototype $\mathbf{P}_n^{i,y}$ to F_n^i with the L2 distance between F_n^i and $P_{j,n}^y$ as follows:

$$\mathbf{P}_n^{i,y} = \arg \min_j \|F_n^i - \mathbf{P}_{j,n}^y\|_2. \quad (1)$$

After obtaining the prototypes at the n th scale, in spirited by [21], we calculate the residuals between F_n^i and $\mathbf{P}_n^{i,y}$ to help the model learn better anomalous features by the equation:

$$D_n^i = F_n^i - \mathbf{P}_n^{i,y}. \quad (2)$$

Additionally, to maintain the richness of anomalous features, we incorporate the original feature map F_n^i and the residual feature map D_n^i together at the respective scales to achieve better anomaly detection and localization as follows:

$$B_n^i = F_n^i \oplus D_n^i. \quad (3)$$

3.2 Scale-Aware Channel Attention Module (SACAM)

We obtain feature maps of different scales from the backbone network and utilize multiscale prototypes and multiscale fusion technologies to obtain the residual feature maps associated with the normal sample distribution at different scales. These residual feature maps serve as inputs for SACAM, thereby providing crucial information for subsequent anomaly detection and localization. In this process, we categorize the feature maps into scales, forming a feature pyramid $B = \{B_n\}$, where the index i used in Eq. (3) is eliminated if the feature pyramid is extracted from a general input X . Furthermore, we use C_n to denote the number of channels in the feature map B_n at the scale n . To capture

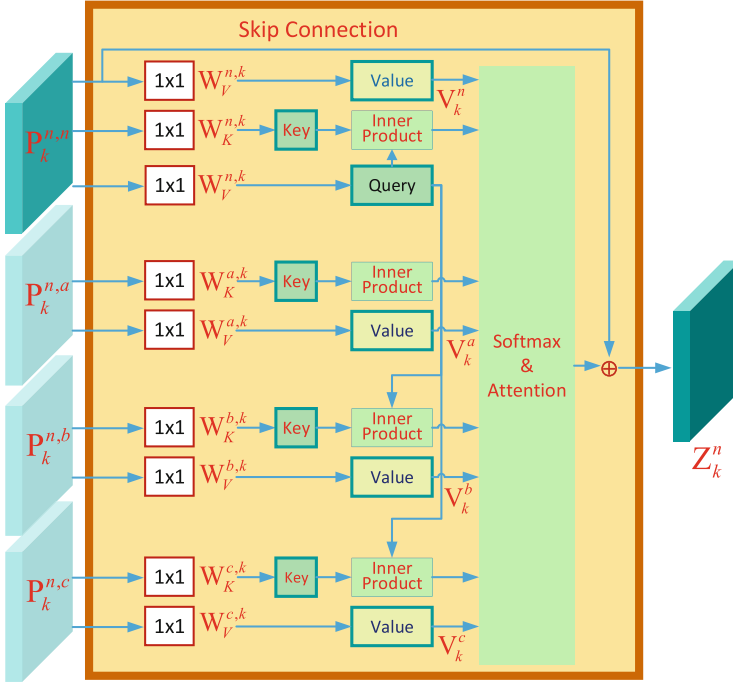


Fig. 3. Detailed operations of the Cross-Attention Block (CAB). The feature map of the n -th scale generates query Q , key K , and value V to interact with the key K and value V of the other three scales (a , b , and c) resulting in cross-scale attention results.

the importance of each channel, we apply a Channel Attention Block (CHAB) to each feature map B_n . As shown in Fig. 2, CHAB utilizes a Global Average Pooling (GAP) operation to capture global information for each channel in B_n , followed by applying the sigmoid activation function to generate a channel attention vector U_n . The dimension of U_n is $1 \times 1 \times C_n$. Subsequently, we adjust all feature maps $\{B_m\}$ to have the same size as B_n , achieved through bilinear interpolation. Then, we perform a 1×1 convolution operation on each adjusted feature map B_m to further extract feature information. Importantly, for each adjusted feature map B_m , its channels are weighted by the previously computed channel attention vector U_n , resulting in a new feature map E_m^n . Thus, we obtain a set of scale-sensitive features $\{E_m^n\}$, which serve as inputs for the subsequent PCSAM module to generate predictions for anomaly detection. This entire process aims to enhance the model's accuracy and robustness in anomaly detection across different scales.

3.3 Patch-Based Cross-Scale Attention Module (PCSAM)

This section introduces a pioneering patch-based cross-scale attention module (PCSAM) to achieve more precise detection of boundaries between anomalies

and normal regions. This module segments the feature map E_m^n into distinct patches, labeled as $\{P_k^{n,m}\}$, $k=1, \dots, \Omega$, where Ω denotes the total patch count, thus enriching the model's ability to understand intricate scenarios. Within this module's architecture, each scale incorporates Ω independent Cross-Attention Blocks (CAB) to facilitate the exchange of information across scales, thereby bolstering anomaly detection performance. As shown in Fig. 3, CAB is designed to enhance the features, ensuring their representativeness and reliability. To maintain the spatial relationship of patches on different scale feature maps, CAB employs the same relative positions of $P_k^{n,m}$ for attention calculation, effectively preserving contextual cues across scales. During cross-scale operations, CAB takes inputs $P_k^{n,n}$, $P_k^{n,a}$, $P_k^{n,b}$, and $P_k^{n,c}$, where a , b , and c denote various scales different from the target scale n . This approach fosters an effective interaction between features at diverse scales, consequently enhancing both the accuracy and the resilience of anomaly detection.

Effective feature extraction is paramount when performing cross-scale attention computation. To achieve this, we rely on three matrices—query, key, and value—as the cornerstone of attention calculation. These matrices are tailored to meet the demands of cross-scale attention computation. The computational procedure of each Cross-Attention Block (CAB) is depicted in Fig. 3. Converting $P_k^{n,n}$ to query Q_k^n , key K_k^n , and value V_k^n involves training three matrices: $W_Q^{n,k}$, $W_K^{n,k}$, and $W_V^{n,k}$. Similarly, for $P_k^{n,a}$, $P_k^{n,b}$, and $P_k^{n,c}$, we train linear matrices $W_K^{a,k}$, $W_V^{a,k}$, $W_K^{b,k}$, $W_V^{b,k}$, $W_K^{c,k}$, and $W_V^{c,k}$ to derive K_k^a , V_k^a , K_k^b , V_k^b , K_k^c , and V_k^c , correspondingly. Leveraging the query Q_k^n , we calculate patch-based cross-scale attention $\beta_k^{n,m}$ between Q_k^n and all keys K_k^m , where $m = n, a, b, c$. The computation process can be summarized as follows:

$$\beta_k^{n,m} = \frac{Q_k^n \otimes K_k^m}{\sum_{m=n,a,b,c} Q_k^n \otimes K_k^m}, \quad (4)$$

where \otimes denotes element-wise multiplication and summation. By utilizing the attention mechanism $\beta_k^{n,m}$, we aggregate the values V_k^m to create a novel feature vector Z_k^n through the following procedure. Furthermore, to tackle the problem of gradient vanishing, we incorporate the original patch feature map $P_k^{n,n}$ via a skip connection.

$$Z_k^n = P_k^{n,n} + \sum_{m=n,a,b,c} \beta_k^{n,m} V_k^m. \quad (5)$$

After acquiring Ω new feature vectors across scales, we merge the patch feature map. This procedure leads to the creation of the feature map Y^n in the following manner:

$$Y^n = \text{Merge}(Z_1^n, \dots, Z_k^n, \dots, Z_\Omega^n), \quad (6)$$

where $\text{Merge}()$ is specifically designed to combine and integrate all the feature vectors Z_k^n together to construct the ultimate feature map Y^n . Subsequently, each Y^n undergoes normalization via bilinear interpolation to ensure uniform size alignment with Y^1 . These normalized feature maps are then concatenated and

processed through a 3×3 convolution operation to create the spill detector, $G : R^{H \times W \times 4C} \rightarrow R^{H \times W \times 2}$, where one channel corresponds to background detection and the other to foreground detection. Finally, the ultimate predicted anomaly probability map M is generated through additional processing as follows:

$$M = G \left(\text{Concatenate} \left(Y^1, Y^2, Y^3, Y^4 \right) \right). \quad (7)$$

A pixel in M is considered part of the foreground if its value in the foreground channel exceeds that of the background channel.

3.4 Loss Function

During the training phase, we optimize our network by utilizing a combination of three different loss functions: cross-entropy loss, dice loss [12], and SSIM loss [19]. The cross-entropy loss measures the negative log-likelihood of the predicted probability distribution across the target labels. It encourages the network to assign higher probabilities to the correct class labels for each pixel in the segmentation mask and is defined as

$$L_{CE} = -\frac{1}{T} \times \sum_{t=1}^T \left(X_t \log(\hat{X}_t) + (1 - X_t) \log(1 - \hat{X}_t) \right), \quad (8)$$

where X_t represents the ground truth class value for a pixel, which is either 1 or 0, indicating whether the corresponding pixel is anomaly or not, T denotes the total number of pixels, and \hat{X}_t represents the predicted value generated by our network. The dice loss evaluates how closely the predicted segmentation mask aligns with the ground truth mask, emphasizing the agreement between them. It effectively penalizes discrepancies, such as false negatives and false positives, thereby capturing fine details in the segmentation output and is given by

$$L_{Dice} = 1 - \frac{\sum_{t=1}^T \hat{X}_t^2 \times X_t^2}{\sum_{t=1}^T \hat{X}_t^2 + X_t^2}. \quad (9)$$

Furthermore, SSIM loss assesses local similarity by comparing small image patches, which complements the global similarity emphasized by the other loss functions. SSIM loss helps preserve intricate features and enhances segmentation accuracy and is mathematically formulated as follows:

$$L_{SSIM} = 1 - \frac{(2\mu_X \mu_{\hat{X}} + C_1) (2\sigma_{X\hat{X}} + C_2)}{(\mu_X^2 + \mu_{\hat{X}}^2 + C_1) (\sigma_X^2 + \sigma_{\hat{X}}^2 + C_2)}, \quad (10)$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation operations respectively. C_1 and C_2 are very small constants included in the denominator to prevent division by zero. Specifically, we set $C_1 = 0.0001$ and $C_2 = 0.0009$. Finally, by integrating Eq. (8)–Eq. (10), we train our PPCA-Net with the following integrated loss function:

$$Loss = L_{CE} + L_{Dice} + L_{SSIM}. \quad (11)$$

3.5 Backbone Details

To ensure effective feature extraction and efficient GPU utilization, we use ResNet18 [7] as the backbone of our PPCA-Net. We optimize computational resources while retaining important features by configuring the channel dimensions of the four-scale feature maps in ResNet18 to 64, 128, 256, and 512, respectively. Furthermore, we adjust the convolutional layers in the last two stages of ResNet18, replacing those with a stride of 2 with convolutions having a stride of 1. This adjustment strikes a balance between computational efficiency and the accuracy of feature extraction. Consequently, we derive four distinct feature maps from the final four stages of ResNet18.

4 Experiments

4.1 Experimental Details

Datasets: To assess the effectiveness of our approach in anomaly detection and localization, we conducted experiments on the MVTec Anomaly Detection (MVTec AD [1]) dataset as a standard benchmark for evaluation. This dataset comprises 10 object sub-datasets and 5 texture sub-datasets, each containing diverse anomalies, facilitating a comprehensive evaluation of surface anomaly detection techniques. Our experiments followed a standard supervised setting [5], where each training set for the subdatasets consists of only 10 abnormal samples. This configuration not only ensures effective model training but also allows for a more accurate assessment of the model’s performance under limited data conditions. By evaluating our method on this dataset, we can thoroughly analyze its performance in industrial defect detection tasks and confirm its suitability and resilience across various types of anomaly.

Evaluation Metrics: Our research methodology involves evaluating outcomes using two metrics: Image-AUROC and Pixel-AUROC. Image-AUROC calculates the average anomaly score from the top 100 scores obtained for each pixel in the output image, compared against the corresponding image label. Pixel-AUROC correlates output scores of each pixel with the labels of the original position pixels. However, Pixel-AUROC will not accurately reflect localization accuracy due to the small fraction of anomalous regions in the entire image, resulting in a significantly higher number of non-anomalous pixels and a consistently low false-positive detection rate. To provide a comprehensive evaluation, we introduced two additional metrics: the Per Region Overlap (PRO) [2] score, treating anomalous regions equally regardless of size, and pixel-level Average Precision (AP) [20], particularly suitable for highly imbalanced classes, such as those in industrial anomaly localization tasks where precision is crucial.

Implementation Details: All images in the experimental datasets are resized to a size of 256×256 . We utilize layer1, layer2, layer3, and layer4 of ResNet-18 pretrained on ImageNet to obtain feature maps of sizes $64 \times 128 \times 128$, $128 \times 64 \times 64$, $256 \times 64 \times 64$, and $512 \times 64 \times 64$, respectively. These sizes are frozen during training. During the training phase, we employ two NVIDIA Tesla V100 GPUs and use the Adam optimizer for parameter optimization. The momentum and weight decay are set to 0.9 and 0.0005, respectively. We initialize the learning rate to 0.005 and decay it by 0.1 every 50 epochs. Each component is trained for 200 epochs with a batch size of 8. Additionally, we apply data augmentation to address the imbalance of abnormal samples during training and better assess the effectiveness of our model. We compute the image-level anomaly score as the average of the top 100 abnormal pixels. PPCA-Net is compared with seven non-supervised SOTA methods and two supervised SOTA methods. The reported results are based on the implementations provided by these methods. The backbones of PatchCore [15], RD4AD [4], CFLOW [6], and CFA [9] are WideResNet50. SSPCAB [14] replaces the second last convolutional layer of the reconstructive encoder in DRAEM [20]. DevNet [13] proposes that the anomaly score output by the network can be further back-propagated to the original image pixels to infer which pixels are classified as the anomaly, used for anomaly localization.

4.2 Anomaly Detection and Localization on MVTec

The outcomes of anomaly detection and localization on MVTec, as shown in Table 1 and Table 2, indicate that our approach achieves comparable or superior performance in both image AUROC (detection) and pixel AUROC (localization) compared to the majority of previous methods. The cross-scale attention mechanism effectively preserves visual features for anomaly detection and localization, spanning from coarse to fine details, regardless of their size. This capability enables better discrimination of anomaly pixels from normal ones, resulting in better detection performance. This suggests that our method effectively detects anomalies and accurately locates them. Thus, in Table 1 and Table 2, our PPCA-Net outperforms other methods in most categories in the “Pixel AUROC” metric. Moreover, our method achieves the best average scores in both both image AUROC (detection) and pixel AUROC (localization).

The Per Region Overlap (PRO) score [2] offers a fair treatment of anomalous regions, irrespective of their size. On the other hand, the pixel-level Average Precision (AP) [20] proves particularly effective in scenarios with highly imbalanced classes, such as those encountered in industrial anomaly localization tasks. To provide a comprehensive evaluation of our anomaly localization capabilities, we have incorporated two additional metrics, PRO and AP, which are presented in Tables 3 and 4. The PRO metric demonstrates that our PPCA-Net outperforms previous unsupervised state-of-the-art (SOTA) models by 4.8% in localizing abnormal regions and surpasses supervised SOTA by 2.6%. This underscores the effectiveness of PPCA-Net in simultaneously identifying abnormal regions of varying sizes. Furthermore, we showcase exceptional anomaly

Table 1. Compared with Unsupervised Anomaly Detection and Localization on MVTec. Best results on Image AUROC or Pixel AUROC are highlighted in bold. (Image AUROC/Pixel AUROC)

Category	Unsupervised						
	CFLOW [6]	DRAEM [20]	SSPCAB [14]	CFA [9]	RD4AD [4]	PatchCore [15]	Ours
Carpet	97.6/ 99.2	96.9/97.5	93.1/92.6	99.9 /98.6	98.7/98.9	99.1/99.0	99.2/99.2
Grid	98.1/98.9	99.9/ 99.7	99.7/99.5	98.6/97.6	100 /98.3	97.3/98.7	100 /99.4
Leather	99.9/ 99.7	100 /99.0	98.7/96.3	100 /99.1	100 /99.4	100 /99.3	100 /99.5
Tile	97.1/96.2	100 /99.2	100 /99.4	99.2/95.1	99.7/95.7	99.3/95.8	100 / 99.7
Wood	98.7/86.0	99.5/95.5	98.4/96.5	100 /94.7	99.5/95.8	99.6/95.1	99.9/ 98.1
Bottle	99.9/97.2	98.0/99.1	95.6/99.2	100 /98.6	100 /98.8	100 /98.6	100 / 99.6
Hazelnut	100 /98.8	100 / 99.7	100 / 99.7	100 /98.6	100 /99.0	100 /98.7	100 /99.3
Metal Nut	98.5/98.6	100 / 99.6	100 /99.4	100 /98.7	100 /97.3	99.9/98.3	100 / 99.8
Pill	96.2/98.9	97.1/97.3	97.4/97.2	97.7/98.0	98.7 /98.1	97.5/97.6	96.5/ 99.1
Toothbrush	98.8/99.0	100 /97.3	97.9/97.3	100 /98.8	100 / 99.1	100 /98.6	99.4/97.5
Transistor	92.9/98.2	91.7/85.2	88.0/84.8	100 /98.1	95.5/92.3	99.9/96.5	98.4/ 98.9
Zipper	97.1/ 99.1	100 / 99.1	100 /98.4	99.5/98.6	97.9/98.3	99.5/98.9	100 / 99.2
Average	97.5/97.7	97.6/96.7	97.1/96.3	99.1/98.0	98.7/97.8	99.2/98.1	99.5 / 99.1

Table 2. Compared with Supervised Anomaly Detection and Localization on MVTec. Best results on Image AUROC or Pixel AUROC are highlighted in bold. (Image AUROC/Pixel AUROC)

Category	Supervised			
	DevNet [13]	DRA [5]	PRN [21]	Ours
Carpet	82.5/97.2	92.5/98.2	99.7 /99.0	99.2/ 99.2
Grid	90.6/87.9	98.6/86.0	99.4/98.4	100 / 99.4
Leather	92.2/94.2	98.9/93.8	100 / 99.7	100 /99.5
Tile	99.9/92.7	100/92.3	100 /99.6	100 / 99.7
Wood	97.9/86.4	99.1/82.9	100 /97.8	99.9/ 98.1
Bottle	99.7/93.9	100/91.3	100 /99.4	100 / 99.6
Hazelnut	99.7/91.1	100/89.6	100 / 99.7	100 /99.3
Metal Nut	98.8/77.8	99.1/79.5	100 /99.7	100 / 99.8
Pill	87.1/82.6	88.3/84.5	99.3 / 99.5	96.5/99.1
Toothbrush	79.2/84.6	87.5/75.5	100 / 99.6	99.4/97.5
Transistor	89.1/56.0	88.3/79.1	99.7 /98.4	98.4/ 98.9
Zipper	99.1/93.7	99.7/96.9	99.7/98.8	100 / 99.2
Average	92.2/85.3	96.1/85.3	99.4/99.0	99.5 / 99.1

localization performance through the more challenging AP metric. Compared to the unsupervised SOTA models, our PPCA-Net outperforms it by 16.6%. As to the supervised SOTA models, it exhibits commendable performance, exceeding

other supervised models by 6.1%. This highlights the exceptional anomaly localization capabilities of our approach. All the compared methods perform poorly in the “Toothbrush” category.

Table 3. Results of the PRO and AP metrics for compared with Unsupervised anomaly localization performance on MVTec. (PRO/AP)

Category	Unsupervised						
	CFLOW [6]	DRAEM [20]	SSPCAB [14]	CFA [9]	RD4AD [4]	PatchCore [15]	Ours
Carpet	97.6/68.3	92.9/65.1	86.4/48.6	93.6/57.2	95.4/56.5	95.5/62.2	99.3/85.2
Grid	96.0/41.2	98.3/62.8	98.0/57.9	92.9/25.8	94.2/15.8	94.0/24.5	99.4/68.8
Leather	99.2/64.5	97.4/72.9	94.0/60.7	95.4/48.5	98.2/47.6	96.9/45.3	99.6/81.7
Tile	89.1/60.1	98.2/95.2	98.1/96.1	83.3/55.9	85.6/54.1	91.3/56.2	99.3/97.5
Wood	82.8/29.0	90.3/74.6	92.8/78.9	85.9/49.0	91.4/48.3	87.1/49.3	98.5/85.9
Bottle	94.0/68.1	96.8/88.9	96.3/89.4	94.6/80.3	96.3/78.0	95.4/76.8	98.7/94.9
Hazelnut	97.1/59.9	98.5/ 92.6	98.2/93.4	95.2/60.0	96.9/60.7	90.9/53.2	99.4/87.6
Metal Nut	91.5/88.0	97.0/97.0	97.7/94.7	91.4/92.2	94.9/78.6	92.6/86.6	98.2/98.9
Pill	95.2/82.0	88.4/47.6	89.6/48.3	95.4/81.9	96.7/76.5	94.5/75.7	98.5/88.4
Toothbrush	95.3/46.3	85.6/45.5	85.5/39.3	86.8/ 55.7	92.3/51.1	94.0/37.9	97.3/51.0
Transistor	82.5/67.5	70.4/39.0	62.5/38.1	95.1/76.2	83.3/54.1	92.3/66.9	98.1/91.4
Zipper	96.6/65.2	96.8/77.6	95.2/76.4	94.3/65.2	95.3/57.5	96.1/62.3	98.5/85.9
Average	93.4/59.6	91.3/68.1	90.8/65.5	92.1/60.0	93.9/55.4	93.9/56.3	98.7/84.7

Table 4. Results of the PRO and AP metrics for compared with Supervised anomaly localization performance on MVTec. (PRO/AP)

Category	Supervised			
	DevNet [13]	DRA [5]	PRN [21]	Ours
Carpet	85.8/45.7	92.2/52.3	97.0/82.0	99.3/85.2
Grid	79.8/25.5	71.5/26.8	95.9/45.7	99.4/68.8
Leather	88.5/8.1	84.0/5.6	99.2/69.7	99.6/81.7
Tile	78.9/52.3	81.5/57.6	98.2/96.5	99.3/97.5
Wood	75.4/25.1	69.7/22.7	95.9/82.6	98.5/85.9
Bottle	83.5/51.5	77.6/41.2	97.0/92.3	98.7/94.9
Hazelnut	83.6/22.1	86.9/22.5	97.4/ 93.8	99.4/87.6
Metal Nut	76.9/35.6	76.7/29.9	95.8/98.0	98.2/98.9
Pill	69.2/14.6	77.0/21.6	97.2/ 91.3	98.5/88.4
Toothbrush	33.5/6.7	56.1/4.5	95.6/ 78.1	97.3/51.0
Transistor	39.1/6.4	49.0/11.0	94.8/85.6	98.1/91.4
Zipper	81.3/19.6	91.0/42.9	95.5/77.6	98.5/85.9
Average	71.4/24.4	73.3/26.0	96.1/78.6	98.7/84.7

4.3 Ablation Study

We proceeded to assess the effectiveness of SACAM and PCSAM separately by integrating them into PPCA-Net and comparing their respective performance outcomes, as outlined in Table 5. Notably, when SACAM is applied in isolation, PPCA-Net already outperforms some previous approaches, underscoring SACAM’s efficacy in improving detection performance. Furthermore, when PCSAM is utilized alone, there are significant enhancements in Image-AUROC, PRO, and AP, further confirming PCSAM’s role in refining subtle detection. The culmination of detection performance occurs when both modules are amalgamated within PPCA-Net, illustrating how SACAM and PCSAM synergize to bolster detection accuracy.

Table 5. Ablation study of PPCA-Net with or w/o SACAM and PCSAM (%).

SACAM	PCSAM	Image-AUROC	Pixel-AUROC	PRO	AP
✓		96.7	97.5	96.5	72.3
	✓	98.1	97.2	97.0	72.8
✓	✓	97.8	98.0	97.5	79.7

4.4 Conclusion

We introduce PPCA-Net as a novel top-performing approach for detecting and locating industrial anomalies. With SACAM, the model can assess channel significance not just within its scale but also across different scales, resulting in a comprehensive and globally attentive feature map. Moreover, PCSAM enables cross-scale analysis, enabling the model to capture information from both large and small anomaly regions simultaneously, thus improving the accuracy of detecting subtle anomalies through a small-patch attention mechanism. Experimental findings validate the efficacy of PPCA-Net in industrial anomaly detection.

References

1. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9592–9600 (2019)
2. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4183–4192 (2020)

3. Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., Steger, C.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint [arXiv:1807.02011](https://arxiv.org/abs/1807.02011) (2018)
4. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9737–9746 (2022)
5. Ding, C., Pang, G., Shen, C.: Catching both gray and black swans: open-set supervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7388–7398 (2022)
6. Gudovskiy, D., Ishizaka, S., Kozuka, K.: CFLOW-AD: real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 98–107 (2022)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Hou, J., Zhang, Y., Zhong, Q., Xie, D., Pu, S., Zhou, H.: Divide-and-assemble: learning block-wise memory for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8791–8800 (2021)
9. Lee, S., Lee, S., Song, B.C.: CFA: coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access* **10**, 78446–78454 (2022)
10. Liu, W., et al.: Towards visually explaining variational autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8642–8651 (2020)
11. Memarsadeghi, N., Mount, D.M., Netanyahu, N.S., le Moigne, J.: A fast implementation of the ISODATA clustering algorithm. *Int. J. Comput. Geom. Appl.* **17**(01), 71–103 (2007)
12. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565–571. IEEE (2016)
13. Pang, G., Shen, C., Van Den Hengel, A.: Deep anomaly detection with deviation networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 353–362 (2019)
14. Ristea, N.C., et al.: Self-supervised predictive convolutional attentive block for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13576–13586 (2022)
15. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14318–14328 (2022)
16. Rudolph, M., Wandt, B., Rosenhahn, B.: Same same but DifferNet: semi-supervised defect detection with normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1907–1916 (2021)
17. Ruff, L., et al.: Deep semi-supervised anomaly detection. arXiv preprint [arXiv:1906.02694](https://arxiv.org/abs/1906.02694) (2019)
18. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. *Neural Comput.* **13**(7), 1443–1471 (2001)
19. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)

20. Zavrtanik, V., Kristan, M., Skočaj, D.: DRAEM-a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8330–8339 (2021)
21. Zhang, H., Wu, Z., Wang, Z., Chen, Z., Jiang, Y.G.: Prototypical residual networks for anomaly detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16281–16291 (2023)



Semi-structured Pruning of Graph Convolutional Networks for Skeleton-Based Recognition

Hichem Sahbi^(✉)

Sorbonne University, CNRS, LIP6, 75005 Paris, France
`hichem.sahbi@sorbonne-universite.fr`

Abstract. Deep neural networks (DNNs) are nowadays witnessing a major success in solving many pattern recognition tasks including skeleton-based classification. The deployment of DNNs on edge-devices, endowed with limited time and memory resources, requires designing lightweight and efficient variants of these networks. Pruning is one of the lightweight network design techniques that operate by removing unnecessary network parts, in a structured or an unstructured manner, including individual weights, neurons or even entire channels. Nonetheless, structured and unstructured pruning methods, when applied separately, may either be inefficient or ineffective.

In this paper, we devise a novel semi-structured method that discards the downsides of structured and unstructured pruning while gathering their upsides to some extent. The proposed solution is based on a differentiable cascaded parametrization which combines (i) a band-stop mechanism that prunes weights depending on their magnitudes, (ii) a weight-sharing parametrization that prunes connections either individually or group-wise, and (iii) a gating mechanism which arbitrates between different group-wise and entry-wise pruning. All these cascaded parametrizations are built upon a common latent tensor which is trained end-to-end by minimizing a classification loss and a surrogate tensor rank regularizer. Extensive experiments, conducted on the challenging tasks of action and hand-gesture recognition, show the clear advantage of our proposed semi-structured pruning approach against both structured and unstructured pruning, when taken separately, as well as the related work.

Keywords: Structured and unstructured pruning · Semi-structured pruning · Graph-convolutional networks · Skeleton-based recognition

1 Introduction

Deep neural networks (DNNs) are nowadays becoming a hotspot in machine learning with increasingly performant models used to approach eclectic pattern recognition tasks [1, 35, 67]. These models are also steadily oversized and this makes their deployment on cheap devices, endowed with limited hardware resources, very challenging. In particular, hand-gesture recognition and

human computer interaction tasks require fast and lightweight DNNs with high recognition performances. However, DNNs are currently showing some saturated improvement in accuracy while their computational efficiency remains a major issue. Among these DNN models, graph convolutional networks (GCNs) are deemed effective especially on non-euclidean domains including skeleton-data [48]. Two families of GCNs exist in the literature: spectral and spatial. Spectral methods project graph signals from the input to the Fourier domain, achieve convolution, prior to back-project the convolved signals in the input domain [3, 4, 23, 71]. Spatial methods proceed differently by aggregating signals through neighboring nodes, using multi-head attention, prior to achieve convolutions (as inner products) on the resulting node aggregates [6, 7, 13, 39, 42]. Spatial GCNs are known to be more effective compared to spectral ones. Nonetheless, with multi-head attention, spatial GCNs become oversized, computationally overwhelming, and their deployment of cheap devices requires making them lightweight and still effective [2, 51].

Several existing works address the issue of lightweight network design, including tensor decomposition [17], quantization [24], distillation [8, 64], neural architecture search [70] and pruning [25–27, 32]. Pruning methods are particularly effective, and their general recipe consists in removing connections in order to enable reduced storage and faster inference with a minimal impact on classification performances. One of the mainstream methods is magnitude pruning (MP) [24] which removes the smallest weight connections before retraining the pruned networks. Two categories of MP techniques exist in the literature: unstructured [24, 25] and structured [15, 41]. Unstructured methods remove weights individually by ranking them according to the importance of their magnitudes whilst structured approaches zero-out groups of weights (belonging to entire rows, columns, filters or channels) according to the importance of their *aggregated* magnitudes. Unstructured MP results into more flexible, accurate networks, and allows reaching any fine-grained targeted pruning rate but requires dedicated hardware to actually achieve efficient computation. In contrast, structured MP offers a more practical advantage by making the resulting DNNs compatible with standard hardware for efficient computation. However, this comes at the expense of a reduced classification performance and coarse-grained pruning rates. *In order to fully exhibit the potential of these two pruning categories, a more suitable framework should gather the upsides of both structured and unstructured pruning while discarding their downsides to some extent.*

In this paper, we introduce a novel variational MP approach that leverages both structured and unstructured pruning. This method dubbed as *semi-structured* is based on a differentiable cascaded weight parametrization composed of (i) a band-stop mechanism enforcing the prior that the smallest weights should be removed, (ii) a weight-sharing that groups mask entries belonging to the same rows, columns, or channels in a given tensor, and (iii) a gating mechanism which arbitrates between different mask group assignments while maximizing the accuracy of the trained lightweight networks. We also consider a budget loss that allows implementing any targeted fine-grained pruning rate and reducing

further the rank of the pruned tensors, resulting into more efficient networks while being closely accurate as shown later in experiments.

2 Related Work

The following review discusses the related work in variational pruning and skeleton-based recognition, highlighting the limitations that motivate our contributions.

Variational Pruning. The general concept behind variational pruning is to learn weights and binary masks that capture the topology of pruned networks. This is achieved by minimizing a global loss that combines a classification error and a regularizer that controls the sparsity (or the cost) of the resulting networks [14–16]. However, these approaches are powerless to implement any given targeted pruning rate without overtrying multiple weighting of the regularizers. Alternative methods explicitly model the network cost using ℓ_0 -based criteria [16, 53] in order to minimize the discrepancy between the observed and the targeted costs. Existing solutions rely on sampling heuristics or relaxation, which promote sparsity—using different regularizers (ℓ_1/ℓ_2 -based, entropy, etc.) [9–12]—but are powerless to implement any given targeted cost exactly and result in overpruning effects leading to disconnected subnetworks. Furthermore, most of the existing solutions, including magnitude pruning [24], decouple the training of network topology from weights, making the learning of pruned networks suboptimal. On another hand, existing pruning methods are either structured [15, 41] or unstructured [24, 25] so their benefit is not fully explored. In contrast to the aforementioned related work, our contribution in this paper seeks to leverage the advantage of both structured and unstructured pruning where the training of masks and weights are coupled on top of shared latent parameters.

Skeleton-Based Recognition. This task has gained increasing interest due to the emergence of sensors like Intel RealSense and Microsoft Kinect. Early methods for hand-gesture and action recognition used RGB [5, 18, 28, 59], depth [52], shape / normals [47, 57, 60–63, 65, 66], and skeleton-based techniques [50]. These methods were based on modeling human motions using handcrafted features [58], dynamic time warping [55], temporal information [20, 68], and temporal pyramids [48]. However, with the resurgence of deep learning, these methods have been quickly overtaken by 2D/3D Convolutional Neural Networks (CNNs) [19, 22, 69], Recurrent Neural Networks (RNNs) [43–46, 48, 49], manifold learning [34, 36–38], attention-based networks [54], and GCNs [29–31, 33]. The recent emergence of GCNs, in particular, has led to their increased use in skeleton-based recognition [4]. These models capture spatial and temporal attention among skeleton joints with better interpretability. However, when tasks involve relatively large input graphs, GCNs (particularly with multi-head attention) become computationally inefficient and require lightweight design techniques. In this paper, we design efficient GCNs that make skeleton-based recognition highly efficient while also being effective.

3 A Glimpse on Graph Convolutional Networks

Let $\mathcal{S} = \{\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)\}_i$ denote a collection of graphs with $\mathcal{V}_i, \mathcal{E}_i$ being respectively the nodes and the edges of \mathcal{G}_i . Each graph \mathcal{G}_i (denoted for short as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$) is endowed with a signal $\{\phi(u) \in \mathbb{R}^s : u \in \mathcal{V}\}$ and associated with an adjacency matrix \mathbf{A} . GCNs aim at learning a set of C filters \mathcal{F} that define convolution on n nodes of \mathcal{G} (with $n = |\mathcal{V}|$) as $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}} = f(\mathbf{A} \mathbf{U}^\top \mathbf{W})$, here $^\top$ stands for transpose, $\mathbf{U} \in \mathbb{R}^{s \times n}$ is the graph signal, $\mathbf{W} \in \mathbb{R}^{s \times C}$ is the matrix of convolutional parameters corresponding to the C filters and $f(\cdot)$ is a nonlinear activation applied entry-wise. In $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}}$, the input signal \mathbf{U} is projected using \mathbf{A} and this provides for each node u , the aggregate set of its neighbors. Entries of \mathbf{A} could be handcrafted or learned so $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}}$ corresponds to a convolutional block with two layers; the first one aggregates signals in $\mathcal{N}(\mathcal{V})$ (sets of node neighbors) by multiplying \mathbf{U} with \mathbf{A} while the second layer achieves convolution by multiplying the resulting aggregates with the C filters in \mathbf{W} . Learning multiple adjacency (also referred to as attention) matrices (denoted as $\{\mathbf{A}^k\}_{k=1}^K$) allows us to capture different contexts and graph topologies when achieving aggregation and convolution. With multiple matrices $\{\mathbf{A}^k\}_k$ (and associated convolutional filter parameters $\{\mathbf{W}^k\}_k$), $(\mathcal{G} \star \mathcal{F})_{\mathcal{V}}$ is updated as $f(\sum_{k=1}^K \mathbf{A}^k \mathbf{U}^\top \mathbf{W}^k)$. Stacking aggregation and convolutional layers, with multiple matrices $\{\mathbf{A}^k\}_k$, makes GCNs accurate but heavy. We propose, in what follows, a method that makes our networks lightweight and still effective.

4 Proposed Method: Semi-structured Magnitude Pruning

In what follows, we formally subsume a given GCN as a multi-layered neural network g_θ whose weights are defined as $\theta = \{\mathbf{W}^1, \dots, \mathbf{W}^L\}$, being L its depth, $\mathbf{W}^\ell \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ its ℓ^{th} layer weight tensor, and d_ℓ the dimension of ℓ . The output of a given layer ℓ is defined as $\phi^\ell = f_\ell(\mathbf{W}^{\ell^\top} \phi^{\ell-1})$, $\ell \in \{2, \dots, L\}$, with f_ℓ an activation function; without a loss of generality, we omit the bias in the definition of ϕ^ℓ .

Pruning consists in zeroing-out a subset of weights in θ by multiplying \mathbf{W}^ℓ with a binary mask $\mathbf{M}^\ell \in \{0, 1\}^{d_{\ell-1} \times d_\ell}$. The binary entries of \mathbf{M}^ℓ are set depending on whether the underlying layer connections are pruned, so $\phi^\ell = f_\ell((\mathbf{M}^\ell \odot \mathbf{W}^\ell)^\top \phi^{\ell-1})$, here \odot stands for the element-wise matrix product. In our definition of semi-structured pruning, entries of the tensor $\{\mathbf{M}^\ell\}_\ell$ are set depending on the prominence and also on how the underlying connections in g_θ are grouped; pruning that removes the entire connections individually (resp. jointly) is referred to as *unstructured* (resp. *structured*) whereas pruning that removes some connections independently and others jointly is dubbed as *semi-structured*. However, such pruning (with $\{\mathbf{M}^\ell\}_\ell$) suffers from several drawbacks. In the one hand, optimizing the discrete set of variables $\{\mathbf{M}^\ell\}_\ell$ is deemed highly combinatorial and intractable especially on large networks. In the other hand, the total number of parameters $\{\mathbf{M}^\ell\}_\ell, \{\mathbf{W}^\ell\}_\ell$ is twice the number of connections in g_θ and this increases training complexity and may also lead to overfitting.

4.1 Semi-structured Weight Parametrization

In order to overcome the aforementioned issues, we consider an alternative *parametrization* that allows finding both the topology of the pruned networks together with their weights, without doubling the size of the training parameters, while making magnitude pruning semi-structured and learning still effective. This parametrization corresponds to the Hadamard product involving a weight tensor and a *cascaded* function applied to the same tensor as

$$\mathbf{W}^\ell = \hat{\mathbf{W}}^\ell \odot [\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)], \quad (1)$$

being $\hat{\mathbf{W}}^\ell$ a latent tensor and $\psi(\hat{\mathbf{W}}^\ell)$ (with $\psi = \psi_3 \circ \psi_2 \circ \psi_1$) a continuous relaxation of \mathbf{M}^ℓ which enforces the prior that (i) smallest weights $\hat{\mathbf{W}}^\ell$ should be removed from the network, (ii) the underlying mask entries $\psi(\hat{\mathbf{W}}^\ell)$ are shared (across tensor rows, columns, channels, etc.) when pruning is structured, and (iii) any given mask entry in $\psi(\hat{\mathbf{W}}^\ell)$ is either unstructurally or structurally pruned. In what follows, we detail the different parametrizations used to define $\psi(\hat{\mathbf{W}}^\ell)$; unless explicitly mentioned, we omit ℓ in the definition of $\hat{\mathbf{W}}^\ell$ and we rewrite it simply as $\hat{\mathbf{W}}$.

Band-stop Parametrization (ψ_1). This parametrization ψ_1 is entry-wise applied to the tensor $\hat{\mathbf{W}}$ and enforces the prior that smallest weights should be removed from the network. In order to achieve this goal, ψ_1 must be (i) bounded in $[0, 1]$, (ii) differentiable, (iii) symmetric, and (iv) $\psi_1(\omega) \rightsquigarrow 1$ when $|\omega|$ is sufficiently large and $\psi_1(\omega) \rightsquigarrow 0$ otherwise. The first and the fourth properties ensure that the parametrization is neither acting as a scaling factor greater than one nor changing the sign of the latent weight, and also acts as the identity for sufficiently large weights, and as a contraction factor for small ones. The second property is necessary to ensure that ψ_1 has computable gradient while the third condition guarantees that only the magnitudes of the latent weights matter. A choice, used in practice, that satisfies these four conditions is

$$\psi_1(\omega) = 2(1 + \exp(-\sigma\omega^2))^{-1} - 1, \quad (2)$$

being σ a scaling factor that controls the crispness (binarization) of mask entries in $\psi_1(\hat{\mathbf{W}})$. According to Eq. 2, σ controls the smoothness of ψ_1 around the support of the latent weights. This allows implementing an annealed (soft) thresholding function that cuts-off all the connections in smooth and differentiable manner as training of the latent parameters evolves. The asymptotic behavior of ψ_1 —that allows selecting the topology of the pruned subnetworks—is obtained as training reaches the latest epochs, and this makes mask entries, in $\psi_1(\hat{\mathbf{W}})$, crisp and (almost) binary. This mask $\psi_1(\hat{\mathbf{W}})$ (rewritten for short as ψ_1) is used as input to the subsequent parameterizations ψ_2 and ψ_3 as shown below.

Weight-sharing Parametrization (ψ_2). This parametrization ψ_2 implements semi-structured pruning by *tying* mask entries belonging to the same rows, columns or channels in the tensor ψ_1 . More precisely, each mask entry in $\psi_2(\psi_1)$ will either be (i) entry-wise evaluated (dependent only on its underlying weight),

or (ii) shared through multiple latent weights belonging to the same row, column or channel of ψ_1 resulting into the following multi-head parametrization (see Fig. 1)

$$\psi_2(\psi_1) = \begin{cases} \psi_2^u(\psi_1) = \psi_1 & \text{unstructured (entry-wise)} \\ \psi_2^r(\psi_1) = \mathbf{vec}^{-1}(\mathbf{P}_r \mathbf{vec}(\psi_1)) & \text{structured (row-wise)} \\ \psi_2^c(\psi_1) = \mathbf{vec}^{-1}(\mathbf{vec}(\psi_1)^\top \mathbf{P}_c) & \text{structured (column-wise)} \\ \psi_2^b(\psi_1) = \mathbf{vec}^{-1}(\mathbf{P}_r \mathbf{P}_c^\top \mathbf{vec}(\psi_1)) & \text{structured (block/channel-wise),} \end{cases} \quad (3)$$

here \mathbf{vec} (resp. \mathbf{vec}^{-1}) reshapes a matrix into a vector (resp. vice-versa), and $\mathbf{P}_r \in \{0, 1\}^{(d_{\ell-1} \times d_\ell)^2}$, $\mathbf{P}_c \in \{0, 1\}^{(d_{\ell-1} \times d_\ell)^2}$ are two adjacency matrices that model the neighborhood system across respectively the rows and the columns of ψ_1 whilst $\mathbf{P}_r \mathbf{P}_c^\top \in \{0, 1\}^{(d_{\ell-1} \times d_\ell)^2}$ models this neighborhood through blocks/channels of ψ_1 . When composed (with ψ_1), the mask ψ_2 inherits all the aforementioned fourth properties: mask entries in $\psi_2(\psi_1)$ remain bounded in $[0, 1]$, differentiable, symmetric, and close to 1 when entries of the latent tensor $\tilde{\mathbf{W}}$ (i.e., inputs of ψ_1) are sufficiently large and 0 otherwise.

Gating Parametrization (ψ_3). As each connection in g_θ is endowed with a multi-head parametrization ψ_2 , we define ψ_3 as a gating mechanism that selects only one of them. More precisely, each mask entry can either be (i) entry-wise pruned, i.e., untied, or (ii) tied to its row, column or block/channel. Again with ψ_3 , the composed parametrization $\psi_3(\psi_2)$ is bounded in $[0, 1]$, differentiable, symmetric and reaches 1 if the entries of the latent tensor $\tilde{\mathbf{W}}$ are sufficiently large, and 0 otherwise. Formally, the gating mechanism ψ_3 is defined as

$$\psi_3(\psi_2) = \underbrace{\psi_2^b}_{\text{block-wise}} + \underbrace{(\bar{\psi}_2^b \odot \psi_2^c)}_{\text{column-wise}} + \underbrace{(\bar{\psi}_2^b \odot \bar{\psi}_2^c) \odot \psi_2^r}_{\text{row-wise}} + \underbrace{(\bar{\psi}_2^b \odot \bar{\psi}_2^c \odot \bar{\psi}_2^r) \odot \psi_2^u}_{\text{entry-wise}}, \quad (4)$$

being $\bar{\psi}_2^b = \mathbf{U} - \psi_2^b$, and \mathbf{U} a tensor of ones with the same dimensions as ψ_2^b (and $\bar{\psi}_2^r$, $\bar{\psi}_2^c$, $\bar{\psi}_2^u$ are similarly defined). It is easy to see that when entries in ψ_1 (and hence ψ_2) are crisp, at most one of these four terms is activated (i.e., equal to one) for each connection in g_θ . From Eq. 4, block-wise pruning has the highest priority, followed by column-wise, row-wise and then entry-wise pruning respectively. This priority allows designing highly efficient lightweight networks with a coarse-granularity budget implementation for block/column/row-wise (structured) pruning, while entry-wise (unstructured) pruning is less computationally efficient but allows reaching any targeted budget with a finer granularity, and thereby with a better accuracy. Note that this parametrization acts a weight regularizer which not only improves the lightweightness of the pruned networks but also their generalization performances (as shown later in experiments). Note also that ψ_1 and ψ_2 are commutable in the cascaded parameterization $\psi = \psi_3 \circ \psi_2 \circ \psi_1$ but ψ_3 should be applied at the end of the cascade.

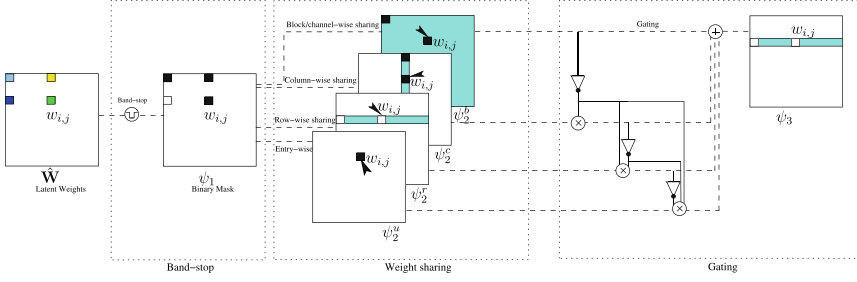


Fig. 1. This figure shows the three stages of the cascaded parametrization including (i) band-stop, (ii) weight-sharing and (iii) gating. Cyan stands for shared connections, and the triangle for the “not gate” operator. For ease of visualization, only 4 connections are shown during the whole evaluation of the parameterization, and only the outcome (1 or 0) of $w_{i,j}$ is shown in the final mask tensor. (Color figure online)

4.2 Budget-Aware Variational Pruning

By considering Eq. 1, we define our semi-structured pruning loss as

$$\mathcal{L}_e(\{\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell) \odot \hat{\mathbf{W}}^\ell\}_\ell) + \lambda \left(\sum_{\ell=1}^{L-1} \mathbf{1}_{d_\ell}^\top [\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)] \mathbf{1}_{d_{\ell+1}} - c \right)^2, \quad (5)$$

being $\mathbf{1}_{d_\ell}$ a vector of d_ℓ ones and the left-hand side term is the cross entropy loss that measures the discrepancy between predicted and ground-truth labels. The right-hand side term is a budget loss that allows reaching any targeted pruning cost c . Nonetheless, it’s worth noticing that actual efficiency is not only related to the pruning rate but also to the actual dimensionality of the tensors. In order to take full advantage of the semi-structured setting of our method, we complement the aforementioned budget function with another one that minimizes the rank of the pruned tensors $\{\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)\}_\ell$. However, as the rank is not differentiable, we consider a surrogate function (as an upper bound) of the rank. Hence, Eq. 5 becomes

$$\begin{aligned} \mathcal{L}_e(\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell) \odot \{\hat{\mathbf{W}}^\ell\}_\ell) + \lambda \left(\sum_{\ell=1}^{L-1} \mathbf{1}_{d_\ell}^\top [\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)] \mathbf{1}_{d_{\ell+1}} - c \right)^2 \\ + \beta \sum_{\ell=1}^{L-1} r[(\psi_3 \circ \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell))], \end{aligned} \quad (6)$$

here $r[\mathbf{W}]$ is a surrogate differentiable rank function set in practice to

$$r[\mathbf{W}] = [\mathbf{1}_{d_{\ell+1}}^\top - \exp(-\gamma \mathbf{1}_{d_\ell}^\top \mathbf{W})] \mathbf{1}_{d_{\ell+1}} + \mathbf{1}_{d_\ell}^\top [\mathbf{1}_{d_\ell} - \exp(-\gamma \mathbf{W} \mathbf{1}_{d_{\ell+1}})], \quad (7)$$

being γ an annealed temperature and $\exp(\cdot)$ is entry-wise applied. Equation 7 seeks to minimize the number of non-null rows/columns in a given tensor \mathbf{W} , and

this allows achieving higher speedup compared to when *only* the budget loss is minimized (see experiments). In Eq. 6, β controls the “structureness” of pruning; large β favors stringent tensors first through blocks, columns and then through rows, while smaller β leads to *mixed* structured and unstructured pruning. Once the above loss optimized, actual rank minimization requires reordering dimensions layer-wise in order to fully benefit from compact tensors and eliminate fragmentation; this is achievable as only outward connections, from unpruned neurons in each layer, are actually pruned during optimization.

Table 1. Jacobians of different parametrizations w.r.t. different settings; here $[\mathbf{J}_1]_{ij,pq} = [\partial\psi_1/\partial\tilde{\mathbf{W}}]_{ij,pq}$, $[\mathbf{J}_2^x]_{ij,pq} = [\partial\psi_2^x/\partial\psi_1]_{ij,pq}$ and $[\mathbf{J}_3^x]_{ij,pq} = [\partial\psi_3/\partial\psi_2^x]_{ij,pq}$ with $\mathbf{x} \in \{u, r, c, b\}$; here u , r , c and b stand for entry-wise, row-wise, column-wise and block-wise respectively. It is easy to see that all these Jacobians are extremely sparse and their evaluation is highly efficient. In this table, NA stands for not applicable as the parametrization ψ_1 is necessarily entry-wise.

Entry-wise	Row-wise	Column-wise	Block-wise
$[\mathbf{J}_1]_{ij,pq} = 1_{\{ij=pq\}}\psi_1'(\tilde{\mathbf{W}}_{pq})$	NA	NA	NA
$[\mathbf{J}_2^u]_{ij,pq} = 1_{\{ij=pq\}}$	$[\mathbf{J}_2^r]_{ij,pq} = [\mathbf{P}_r]_{ij,pq}$	$[\mathbf{J}_2^c]_{ij,pq} = [\mathbf{P}'_c]_{ij,pq}$	$[\mathbf{J}_2^b]_{ij,pq} = [\mathbf{P}_r\mathbf{P}'_c]_{ij,pq}$
$[\mathbf{J}_3^u]_{ij,pq} = 1_{\{ij=pq\}}$ $\times [\bar{\psi}_2^b \odot \bar{\psi}_2^c \odot \bar{\psi}_2^r]_{pq}$	$[\mathbf{J}_3^r]_{ij,pq} = 1_{\{ij=pq\}}$ $\times [\bar{\psi}_2^b \odot \bar{\psi}_2^c \odot \bar{\psi}_2^u]_{pq}$	$[\mathbf{J}_3^c]_{ij,pq} = 1_{\{ij=pq\}}$ $\times [\bar{\psi}_2^b \odot \bar{\psi}_2^r \odot \bar{\psi}_2^u]_{pq}$	$[\mathbf{J}_3^b]_{ij,pq} = 1_{\{ij=pq\}}$ $\times [\bar{\psi}_2^c \odot \bar{\psi}_2^r \odot \bar{\psi}_2^u]_{pq}$

4.3 Optimization

Let \mathcal{L} denote the global loss in Eq. 6, the update of $\{\hat{\mathbf{W}}^\ell\}_\ell$ is achieved using the gradient of \mathcal{L} obtained by propagating the gradients through g_θ . More precisely, considering the parametrization in Eq. 1, the gradient of the global loss w.r.t. $\hat{\mathbf{W}}^\ell$ is obtained as

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{W}}^\ell} = \frac{\partial \mathcal{L}}{\partial \psi(\hat{\mathbf{W}}^\ell)} \frac{\partial \psi(\hat{\mathbf{W}}^\ell)}{\partial \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)} \frac{\partial \psi_2 \circ \psi_1(\hat{\mathbf{W}}^\ell)}{\partial \psi_1(\hat{\mathbf{W}}^\ell)} \frac{\partial \psi_1(\hat{\mathbf{W}}^\ell)}{\partial \hat{\mathbf{W}}^\ell}, \quad (8)$$

here the original gradient $\partial \mathcal{L} / \partial \psi(\hat{\mathbf{W}}^\ell)$ is obtained from layer-wise backpropagation, and $\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{W}}^\ell}$ is obtained by multiplying the original gradient by the three rightmost Jacobians (whose matrix forms are shown in Table 1).

In the above objective function (Eq. 6), $\beta = 0.1$ and λ is overestimated (to 1000 in practice) in order to make Eq. 6 focusing on the implementation of the budget. As training reaches its final epochs, the budget loss reaches its minimum and the gradient of the global objective function will be dominated by the gradient of \mathcal{L}_e (and to some extent by the gradient of the surrogate rank function); this allows improving both classification performances and efficiency as shown subsequently.

5 Experiments

This section assesses baseline and pruned GCNs’ performance in skeleton-based recognition using SBU Interaction [65] and the First Person Hand Action (FPHA) [21] datasets, comparing our lightweight GCNs against related pruning techniques. SBU is an interaction dataset acquired using the Microsoft Kinect sensor, it contains 282 moving skeleton sequences performed by two interacting individuals and belonging to 8 categories. Each pair of interacting individuals corresponds to two 15 joint skeletons, each one encoded with a sequence of its 3D coordinates across video frames. The evaluation protocol follows the train-test split as in the original dataset release [65]. The FPHA dataset includes 1175 skeletons belonging to 45 action categories performed by 6 different individuals in 3 scenarios. Action categories are highly variable, including various styles, speed, scale, and viewpoint. Each skeleton includes 21 hand joints, each one again encoded with a sequence of its 3D coordinates across video frames. The performances of different methods are evaluated using the 1:1 setting proposed in [21] with 600 action sequences for training and 575 for testing. The average accuracy over all classes of actions is reported in all experiments.

Table 2. Comparison of our baseline GCN against related work on the SBU database. Results shown in Bold stand for the best performances while those underlined correspond to the second best performances.

Method	Accuracy (%)
Raw Position [65]	49.7
Joint feature [60]	86.9
CHARM [61]	86.9
H-RNN [43]	80.4
ST-LSTM [44]	88.6
Co-occurrence-LSTM [48]	90.4
STA-LSTM [54]	91.5
ST-LSTM + Trust Gate [44]	93.3
VA-LSTM [46]	97.6
GCA-LSTM [45]	94.9
Riemannian manifold. traj [36]	93.7
DeepGRU [49]	95.7
RHCN + ACSC + STUFE [30]	98.7
Our baseline GCN	<u>98.4</u>

Table 3. Comparison of our baseline GCN against related work on the FPHA database.

Method	Color	Depth	Pose	Accuracy (%)
2-stream-color [19]	✓	✗	✗	61.56
2-stream-flow [19]	✓	✗	✗	69.91
2-stream-all [19]	✓	✗	✗	75.30
HOG2-dep [52]	✗	✓	✗	59.83
HOG2-dep+pose [52]	✗	✓	✓	66.78
HON4D [62]	✗	✓	✗	70.61
Novel View [63]	✗	✓	✗	69.21
1-layer LSTM [48]	✗	✗	✓	78.73
2-layer LSTM [48]	✗	✗	✓	80.14
Moving Pose [66]	✗	✗	✓	56.34
Lie Group [55]	✗	✗	✓	82.69
HBRNN [43]	✗	✗	✓	77.40
Gram Matrix [68]	✗	✗	✓	85.39
TF [20]	✗	✗	✓	80.69
JOULE-color [28]	✓	✗	✗	66.78
JOULE-depth [28]	✗	✓	✗	60.17
JOULE-pose [28]	✗	✗	✓	74.60
JOULE-all [28]	✓	✓	✓	78.78
Huang et al. [34]	✗	✗	✓	84.35
Huang et al. [38]	✗	✗	✓	77.57
HAN [37]	✗	✗	✓	<u>85.74</u>
Our baseline GCN	✗	✗	✓	86.43

Input Graphs. Let’s consider a sequence of skeletons $\{S^t\}_t$ with $S^t = \{\hat{p}_1^t, \dots, \hat{p}_n^t\}$ being the 3D skeleton coordinates at frame t , and $\{\hat{p}_j^t\}_t$ a joint trajectory through successive frames. We define an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a finite collection of trajectories, with each node $v_j \in \mathcal{V}$ in \mathcal{G} being a trajectory $\{\hat{p}_j^t\}_t$, and an edge $(v_j, v_i) \in \mathcal{E}$ exists between two nodes if the underlying

Table 4. This table shows detailed performances and ablation study on SBU for different pruning rates. “none” stands for no-actual speedup is observed as the underlying tensors/architecture remain shaped identically to the unpruned network (despite having pruned connections). For structured, unstructured and semi-structured settings, when “rank optimization” is not used, only pruning rate is considered in the loss together with cross entropy. When “rank optimization” is used, all the three terms are combined in the loss.

Pruning rates	Accuracy (%)	SpeedUp	Observation
0%	98.40	none	Baseline GCN
70%	93.84	none	Band-stop Weight Param.
90%	87.69	426×	Structured
	89.23	487×	Structured (+ rank optimization)
	93.84	none	Unstructured
	93.84	16×	Unstructured (+ rank optimization)
	<u>90.76</u>	40×	Semi-structured
	<u>89.23</u>	52×	Semi-structured (+ rank optimization)
95%	87.69	678×	Structured
	87.69	787×	Structured (+ rank optimization)
	92.30	none	Unstructured
	<u>92.30</u>	16×	Unstructured (+ rank optimization)
	92.30	<u>109×</u>	Semi-structured
	93.84	106×	Semi-structured (+ rank optimization)
98%	81.53	797×	Structured
	81.53	2195×	Structured (+ rank optimization)
	89.23	none	Unstructured
	89.23	106×	Unstructured (+ rank optimization)
	<u>83.07</u>	<u>135×</u>	Semi-structured
	<u>86.15</u>	<u>607×</u>	Semi-structured (+ rank optimization)
Comparative (regularization-based) pruning			
98%	55.38	none	MP+ ℓ_0 -reg.
	73.84	none	MP+ ℓ_1 -reg.
	61.53	none	MP+Entropy-reg.
	75.38	none	MP+Cost-aware-reg.

trajectories are spatially neighbors. Each trajectory is processed using *temporal chunking*, which splits the total duration of a sequence into M evenly-sized temporal chunks (with $M = 4$ in practice). Then, joint coordinates $\{\hat{p}_j^t\}_t$ of the trajectory are assigned to these chunks, based on their time stamps. The averages of these chunks are concatenated in order to create the raw description of the trajectory (denoted as $\phi(v_j) \in \mathbb{R}^s$ with $s = 3 \times M$). This process preserves the temporal structure of trajectories while being frame-rate and duration agnostic.

Implementation Details and Baseline GCNs. All the GCNs have been trained using the Adam optimizer for 2,700 epochs with a batch size of 200 for SBU and 600 for FPHA, a momentum of 0.9, and a global learning rate (denoted as $\nu(t)$) inversely proportional to the speed of change of the loss used to train the networks; with $\nu(t)$ decreasing as $\nu(t) \leftarrow \nu(t-1) \times 0.99$ (resp. increasing as $\nu(t) \leftarrow \nu(t-1)/0.99$) when the speed of change of the loss in Eq. 6 increases (resp. decreases). Experiments were run on a GeForce GTX 1070 GPU device with 8 GB memory, without dropout or data augmentation. The baseline GCN architecture for SBU includes an attention layer of 8 heads, a convolutional layer of 16 filters, a dense fully connected layer, and a softmax layer. The baseline GCN

Table 5. This table shows detailed performances and ablation study on FPHA for different pruning rates. “none” stands for no-actual speedup is observed as the underlying tensors/architecture remain shaped identically to the unpruned network (despite having pruned connections).

Pruning rates	Accuracy (%)	SpeedUp	Observation
0%	86.43	none	Baseline GCN
50%	85.56	none	Band-stop Weight Param.
90%	68.00	274 ×	Structured
	71.30	547 ×	Structured (+ rank optimization)
	83.82	none	Unstructured
	84.17	16×	Unstructured (+ rank optimization)
	<u>78.60</u>	33 ×	Semi-structured
	<u>80.52</u>	38 ×	Semi-structured (+ rank optimization)
95%	56.69	759 ×	Structured
	62.60	931 ×	Structured (+ rank optimization)
	78.78	none	Unstructured
	80.17	29×	Unstructured (+ rank optimization)
	<u>72.17</u>	197 ×	Semi-structured
	<u>74.60</u>	214 ×	Semi-structured (+ rank optimization)
98%	47.47	1479 ×	Structured
	49.04	1399 ×	Structured (+ rank optimization)
	78.08	none	Unstructured
	77.56	126×	Unstructured (+ rank optimization)
	<u>75.13</u>	33 ×	Semi-structured
	<u>73.91</u>	278 ×	Semi-structured (+ rank optimization)
Comparative (regularization-based) pruning			
98%	64.69	none	MP+ ℓ_0 -reg.
	70.78	none	MP+ ℓ_1 -reg.
	67.47	none	MP+Entropy-reg.
	69.91	none	MP+Cost-aware-reg.

architecture for FPHA is heavier and includes 16 heads, a convolutional layer of 32 filters, a dense fully connected layer, and a softmax layer. Both the baseline GCN architectures, on the SBU and the FPHA benchmarks, are accurate (see Tables 2 and 3), and our goal is to make them lightweight while maintaining their high accuracy.

Performances, Comparison and Ablation. Tables 4 and 5 show a comparison and an ablation study of our method both on the SBU and the FPHA datasets. First, according to the observed results, when only the cross entropy loss is used without budget (i.e., $\lambda = \beta = 0$ in Eq. 6), performances are close to the initial heavy GCNs (particularly on FPHA), with less parameters¹ as this produces a regularization effect similar to [56]. Then, when pruning is structured, the accuracy is relatively low but the speedup is important particularly for high pruning regimes. When pruning is unstructured, the accuracy reaches its highest value, but no actual speedup is observed as the architecture of the pruned networks remains unchanged (i.e., not compact). When pruning is semi-structured, we observe the best trade-off between accuracy and speedup; in other words, *coarsely* pruned parts of the network (related to entire block/column/row

¹ Pruning rate does not exceed 70% and no control on this rate is achievable when $\lambda = 0$.

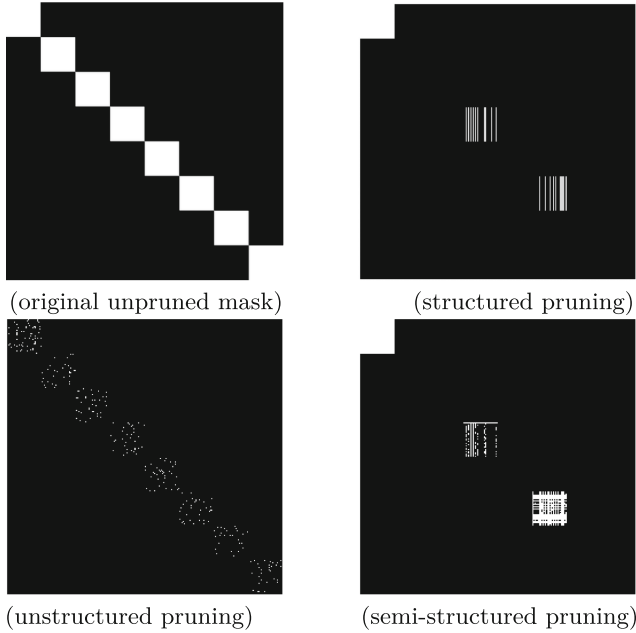


Fig. 2. This figure shows a crop of the mask tensor obtained after the gating parametrization when trained on the FPHA dataset. Top-left corresponds to the original mask (without pruning) while the others correspond to masks obtained with structured, unstructured and semi-structured pruning respectively. In all these masks, each diagonal block corresponds to a channel. Better to zoom the PDF.

connections) lead to high speedup and efficient computation, whereas *finely* pruned parts (related to individual connections) lead to a better accuracy with a contained marginal impact on computation, so speedup is still globally observed with a significant amount.

Extra comparison of our method against other regularizers shows a substantial gain. Indeed, our method is compared against different variational pruning with regularizers plugged in Eq. 6 (instead of our proposed budget and rank regularizers), namely ℓ_0 [16], ℓ_1 [11], entropy [12] and ℓ_2 -based cost [40], all without our parametrization. From the observed results, the impact of our method is substantial for different settings and for equivalent pruning rate (namely 98%). Note that when alternative regularizers are used, multiple settings (trials) of the underlying mixing hyperparameters (in Eq. 6) are considered prior to reach the targeted pruning rate, and this makes the whole training and pruning process overwhelming. While cost-aware regularization makes training more affordable, its downside resides in the observed collapse of trained masks; this is a well known effect that affects performances at high pruning rates. Finally, Fig. 2 shows examples of obtained mask tensors taken from the second (attention) layer of the pruned GCN. For semi-structured pruning, we observe a compact tensor layer

with some individually pruned connections whereas structured and unstructured pruning—when applied separately—either produce *compact* or *spread* tensors, with a negative impact on respectively *accuracy* or *speed*. In sum, semi-structured pruning gathers the advantages of *both* while discarding their inconveniences.

6 Conclusion

This paper introduces a novel magnitude pruning approach that combines *both* the strengths of structured and unstructured pruning methods while discarding their drawbacks. The proposed method, dubbed as *semi-structured*, is based on a novel cascaded weight parametrization including band-stop, weight-sharing, and gating mechanisms. Our pruning method also relies on a budget loss that allows implementing fine-grained targeted pruning rates while also reducing the rank of the pruned tensors resulting in more efficient and still effective networks. Extensive experiments, conducted on the challenging task of skeleton-based recognition, corroborate all these findings.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
2. Sahbi, H.: Phase-field models for lightweight graph convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4643–4649 (2023)
3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
4. Li, R., Wang, S., Zhu, F., Huang, J.: Adaptive graph convolutional neural networks. In: Proceedings of the AAAI, vol. 32 (2018)
5. Wang, L., Sahbi, H.: Directed acyclic graph kernels for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3168–3175 (2013)
6. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: IJCNN, vol. 2, pp. 729–734. IEEE (2005)
7. Knyazev, B., Taylor, G.W., Amer, M.: Understanding attention and generalization in graph neural networks. In: Advances in NIPS, vol. 32 (2019)
8. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. In: Proceedings of the AAAI, vol. 34, pp. 5191–5198 (2020)
9. Carreira-Perpinán, M.A., Idelbayev, Y.: “learning-compression” algorithms for neural net pruning. In: Proceedings of the IEEE CVPR, pp. 8532–8541 (2018)
10. Gordon, A., et al.: MorphNet: fast & simple resource-constrained structure learning of deep networks. In: Proceedings of the IEEE CVPR, pp. 1586–1595 (2018)
11. Koneru, B.N.G., Vasudevan, V.: Sparse artificial neural networks using a novel smoothed lasso penalization. *IEEE Trans. Circuits Syst. II Express Briefs* **66**(5), 848–852 (2019)
12. Wiedemann, S., Marban, A., Müller, K.R., Samek, W.: Entropy-constrained training of deep neural networks. In: IJCNN (2019)

13. Sahbi, H., Audibert, J.-Y., Keriven, R.: Context-dependent kernels for object classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(4), 699–708 (2011)
14. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: *Advances in NIPS*, vol. 29 (2016)
15. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: *Proceedings of the IEEE ICCV*, pp. 2736–2744 (2017)
16. Louizos, C., Welling, M., Kingma, D.P.: Learning sparse neural networks through l_0 regularization. *arXiv preprint [arXiv:1712.01312](https://arxiv.org/abs/1712.01312)* (2017)
17. Howard, A., et al. Searching for mobilenetv3. In: *Proceedings of the IEEE/CVF ICCV*, pp. 1314–1324 (2019)
18. Wang, L., Sahbi, H.: Bags-of-daglets for action recognition. In: *IEEE International Conference on Image Processing (ICIP)*, pp. 1550–1554. IEEE (2014)
19. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: *IEEE CVPR* (2016)
20. Garcia-Hernando, G., Kim, T.K.: Transition forests: Learning discriminative temporal transitions for action recognition and detection. In: *Proceedings of the IEEE CVPR*, pp. 432–440 (2017)
21. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In: *Proceedings of the IEEE CVPR*, pp. 409–419 (2018)
22. Mazari, A., Sahbi, H.: Deep temporal pyramid design for action recognition. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2077–2081. IEEE (2019)
23. Mazari, A., Sahbi, H.: MLGCN: multi-Laplacian graph convolutional networks for human action recognition. In: *The British Machine Vision Conference (BMVC)* (2019)
24. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint [arXiv:1510.00149](https://arxiv.org/abs/1510.00149)* (2015)
25. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: *Advances in NIPS*, vol. 28, (2015)
26. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. In: *Advances in NIPS*, vol. 2 (1989)
27. Hassibi, B. and Stork, D.: Second order derivatives for network pruning: Optimal brain surgeon. In: *Advances in NIPS*, vol. 5 (1992)
28. Hu, J.F., Zheng, W.S., Lai, J., Zhang, J.: Jointly learning heterogeneous features for RGB-D activity recognition. In: *CVPR* (2015)
29. Chaolong, L., Zhen, C., Wenming, Z., Chunyan, X., Jian, Y.: Spatio-temporal graph convolution for skeleton based action recognition. In: *AAAI* (2018)
30. Li, S., Jiang, T., Huang, T. and Tian, Y.: Global co-occurrence feature learning and active coordinate system conversion for skeleton-based action recognition. In: *Proceedings of the IEEE/CVF WACV*, pp. 586–594 (2020)
31. Wen, Y.-H., Gao, L., Hongbo, F., Zhang, F.-L., Xia, S.: Graph CNNs with motif and variable temporal block for skeleton-based action recognition. *Proc. AAAI* **33**, 8989–8996 (2019)
32. Sahbi, H.: Topologically-consistent magnitude pruning for very lightweight graph convolutional networks. In: *IEEE International Conference on Image Processing (ICIP)*, pp. 3495–3499. IEEE (2022)
33. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: *AAAI*, vol. 32 (2018)

34. Huang, Z., Van Gool, L.: A Riemannian network for SPD matrix learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
35. Jiu, M., Sahbi, H.: Deep representation design from deep kernel networks. *Pattern Recogn.* **88**, 447–457 (2019)
36. Kacem, A., Daoudi, M., Amor, B.B., Berretti, S., Alvarez-Paiva, J.C.: A novel geometric framework on gram matrix trajectories for human behavior understanding. *IEEE TPAMI*, **42**(1), 1–14 (2018)
37. Liu, J., Wang, Y., Xiang, S., Pan, C.: HAN: an efficient hierarchical self-attention network for skeleton-based gesture recognition. *arXiv preprint [arXiv:2106.13391](https://arxiv.org/abs/2106.13391)* (2021)
38. Huang, Z., Wu, J., Van Gool, L.: Building deep networks on Grassmann manifolds. In: Proceedings of the AAAI, vol. 32 (2018)
39. Sahbi, H.: Kernel-based graph convolutional networks. In: 25th International Conference on Pattern Recognition (ICPR), pp. 4887–4894. IEEE (2021)
40. Lemaire, C., Achkar, A. and Jodoin, P.M.: Structured pruning of neural networks with budget-aware regularization. In: CVPR, pp. 9108–9116 (2019)
41. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. *arXiv preprint [arXiv:1608.08710](https://arxiv.org/abs/1608.08710)* (2016)
42. Sahbi, H.: Learning connectivity with graph convolutional networks. In: 25th International Conference on Pattern Recognition (ICPR), pp. 9996–10003. IEEE (2021)
43. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: CVPR, pp. 1110–1118 (2015)
44. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal LSTM with trust gates for 3d human action recognition. In: ECCV 2016
45. Liu, J., Wang, G., Duan, L.Y., Abdiyeva, K., Kot, A.C.: Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Trans. Image Process.* **27**(4), 1586–1599 (2017)
46. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: ICCV, pp. 2117–2126 (2017)
47. Sahbi, H.: Kernel PCA for similarity invariant shape recognition. *Neurocomputing* **70**(16–18), 3034–3045 (2007)
48. Zhu, W., et al.: Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In: AAAI, vol. 30 (2016)
49. Maghoumi, M., LaViola, J.J.: DeepGRU: deep gesture recognition utility. In: ISVC 2019
50. Wang, P., Li, W., Ogunbona, P., Wan, J., Escalera, S.: A survey. *CVIU, RGB-d based human motion recognition with deep learning* (2018)
51. Sahbi, H.: Lightweight connectivity in graph convolutional networks for skeleton-based recognition. In: IEEE International Conference on Image Processing (ICIP), pp. 2329–2333. IEEE (2021)
52. Eshed Ohn-Bar and Mohan Manubhai Trivedi: Hand gesture recognition in real time for automotive interfaces: a multimodal vision-based approach and evaluations. *IEEE Trans. Intell. Transp. Syst.* **15**(6), 2368–2377 (2014)
53. Pan, W., Dong, H., Guo, Y.: DropNeuron: simplifying the structure of deep neural networks. *arXiv preprint [arXiv:1606.07326](https://arxiv.org/abs/1606.07326)* (2016)
54. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: AAAI, vol. 31 (2017)
55. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3D skeletons as points in a lie group. In: CVPR (2014)

56. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., Fergus, R.: Regularization of neural networks using DropConnect. In: ICML, pp. 1058–1066. PMLR (2013)
57. Sahbi, H., Fleuret, F.: Kernel methods and scale invariance using the triangular kernel. Technical report, INRIA (2004)
58. Yang, X., Tian, Y.L.: Effective 3D action recognition using EigenJoints. *J. Vis. Commun. Image Represent.* **25**(1), 2–11 (2014)
59. Yuan, F., Xia, G.-S., Sahbi, H., Prinet, V.: Mid-level features and spatio-temporal context for activity recognition. *Pattern Recogn.* **45**(12), 4182–4191 (2012)
60. Ji, Y., Ye, G., Cheng, H.: Interactive body part contrast mining for human interaction recognition. In: ICMEW, pp. 1–6. IEEE (2014)
61. Li, W., Wen, L., Chuah, M.C. and Lyu, S.: A practical recognition system. In: IEEE ICCV, Category-Blind Human Action Recognition (2015)
62. Oreifej, O., Liu, Z.: Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences. In: CVPR, pp. 716–723 (2013)
63. Rahmani, H., Mian, A.: 3D action recognition from novel viewpoints. In: CVPR, pp. 1506–1515 (2016)
64. Sahbi, H., Geman, D., Perona, P.: A hierarchy of support vector machines for pattern detection. *J. Mach. Learn. Res.* **7**(10), 2087–2123 (2006)
65. Yun, K., Honorio, J., Chattopadhyay, D., Berg, T.L., Samaras, D.: Two-person interaction detection using body-pose features and multiple instance learning. In: CVPRW, pp. 28–35. IEEE (2012)
66. Zanfır, M., Leordeanu, M., Sminchisescu, C.: The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection. In: ICCV, pp. 2752–2759 (2013)
67. Jiu, M., Sahbi, H.: Nonlinear deep kernel learning for image annotation. *IEEE Trans. Image Process.* **26**(4), 1820–1832 (2017)
68. Zhang, X., Wang, Y., Gou, M., Sznai, M., Camps, O.: Efficient temporal sequence comparison and classification using gram matrix embeddings on a Riemannian manifold. In: CVPR, pp. 4498–4507 (2016)
69. Nunez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S., Velez, J.F.: Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recogn.* **76**, 80–94 (2018)
70. Li, Y., Zhao, P., Yuan, G., et al.: Pruning-as-search: efficient neural architecture search via channel pruning and structural reparameterization. *arXiv preprint [arXiv:2206.01198](https://arxiv.org/abs/2206.01198)* (2022)
71. Sahbi, H.: Learning laplacians in chebyshev graph convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2064–2075 (2021)



Data Pruning via Separability, Integrity, and Model Uncertainty-Aware Importance Sampling

Steven Grosz¹(✉), Rui Zhao², Rajeep Ranjan², Hongcheng Wang²,
Manoj Aggarwal², Gerard Medioni², and Anil Jain¹

¹ Michigan State University, East Lansing, MI 48824, USA
{groszste,jain}@msu.edu

² Amazon, Seattle, WA 98101, USA
{zhaori,rvrnanjan,hongchw,manojagg,medioni}@amazon.com

Abstract. This paper improves upon existing data pruning methods for image classification by introducing a novel pruning metric and pruning procedure based on importance sampling. The proposed pruning metric explicitly accounts for data separability, data integrity, and model uncertainty, while the sampling procedure is adaptive to the pruning ratio and considers both intra-class and inter-class separation to further enhance the effectiveness of pruning. Furthermore, the sampling method can readily be applied to other pruning metrics to improve their performance. Overall, the proposed approach scales well to high pruning ratio and generalizes better across different classification models, as demonstrated by experiments on four benchmark datasets, including the fine-grained classification scenario.

Keyword: Machine Learning, Data Pruning, Coreset Selection.

1 Introduction

The escalating size of models and datasets has led to an increase in the cost of training deep models. To address the challenges in maintaining high accuracy and scalability in the training process, various data reduction strategies have been proposed. These strategies aim to curate a smaller set of data that maximally retains information of the original dataset, facilitating more efficient learning of feature representations. Two overarching categories encapsulate most data reduction techniques: data pruning and data distillation. Data pruning involves identifying a subset of existing data for retention, whereas data distillation focuses on synthesizing a small number of new samples that lead to similar model accuracy as the original dataset [27].

S. Grosz—This author’s contribution was performed as an intern with Amazon.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78166-7_26.

While many data pruning methods are effective, there are remaining challenges that hinder the practical application of these techniques. One limitation of the existing pruning methods is the lack of robustness to data noise. For example, one proposed method utilizes the l_2 error of the predictions [16] as a pruning metric, which is unable to accurately discriminate between difficult but useful samples and noisy, unusable samples [9]. This is due to the absence of a measure of sample quality, in addition to sample difficulty. Another drawback lies in the potential exacerbation of class imbalances by existing pruning strategies. The strict sampling approach, based on an increasing order of sample difficulty, may inadvertently lead to or intensify existing disparities between classes [17]. While some prior methods suggest pruning an uneven number of samples per majority and minority classes to balance the number of samples [23], such approaches overlook the inherent variations in difficulty among different classes. Moreover, the adaptability of existing methods is questionable, particularly concerning the decision of whether to prune difficult or easy samples. This decision hinges on factors such as the initial data volume and the chosen pruning ratio. The prevailing observation, as highlighted in [20], underscores the optimal strategy of retaining hard samples when the dataset is large and easy samples when it is small. To the best of our knowledge, few prior methods have embraced this insight in their pruning procedures.

In this paper, we aim to design a scalable data pruning metric which improves upon the state-of-the-art by overcoming the previously mentioned limitations. Concretely, our contributions towards this goal are the following:

1. A novel data pruning metric that intuitively captures multiple factors of data utility, including **S**eparability, **I**ntegrity, and **M**odel uncertainty (SIM).
2. An importance sampling procedure to prune data, which can be used with commonly used pruning metrics to improve the effectiveness of pruned data.
3. Evaluation on four benchmark datasets ranging from small to large number of classes show our method, which combines SIM with **S**ampling *i.e.*, SIMS, has better scalability when the pruning ratios is high, better cross-model generalization, and reduced time needed to calculate pruning metrics than other approaches.

2 Related Work

2.1 Data Pruning

One of the first pruning approaches proposed was Forgetting Scores [25], where samples were pruned based on the number of times they transitioned from being correctly to incorrectly classified throughout the training process of an expert model trained on the full dataset. Although performing well across many different classification datasets, Forgetting Scores suffer from high computational costs due to training the expert models to convergence on the full training dataset. To reduce this time spent computing pruning scores, Paul *et al.* [16] proposed a new metric, EL2N, based on the magnitude of the error norm averaged across

multiple expert models trained on the full dataset. Importantly, the authors showed that good performance could be achieved through training the expert models for only 20 epochs on the original datasets. However, a limitation present in both EL2N and Forgetting Scores is that they do not explicitly consider the recognizability of those samples in relation to their class centers in the embedding space. Similarly, other approaches have proposed using other techniques *e.g.*, gradient of the loss [7, 8], generalization influence [29], label noise [15], moving-one-sample-out [24], *etc.* But these methods also do not consider the inherent separability of different classes.

To address this limitation, Sorscher *et al.* [20] proposed Prototype and Self-Prototype scores for pruning. Here, the distance of each data sample from its corresponding class center is used as a pruning score, where samples farther from the class center are considered more challenging and are likely to be retained. With these simple metrics, the authors obtained similar pruning performance as the baseline EL2N and Forgetting Scores. However, two important limitations remain. First, neither of the previously mentioned pruning metrics explicitly models the quality of each data sample. This is important as without a measure of sample quality, the distinction between difficult, recognizable samples, and difficult, unrecognizable samples is missed. Secondly, even though Prototype scores consider the recognizability of each data sample in relation to its own class center, they do not utilize the separability of samples compared to other class samples. Thus, information regarding the inherent separability of classes is overlooked. Our proposed metric aims to address each of the aforementioned challenges by explicitly measuring indications of data integrity, model uncertainty, and class separability in computing scores for pruning.

Another importance consideration in data pruning is not only the metric used to obtain pruning scores, but in how those scores are used to sample the original data for pruning. Many approaches sort the scores from low to high and retain the percentage of samples based on the highest scores. However, as noted in [20], the optimal sampling strategy depends on the amount of data retained. With a similar motivation, Xia *et al.* [28] proposed to sample scores in proximity to the median of each data class’s distribution and showed improved results compared to using a strict sorting, especially at higher pruning ratios. Zheng *et al.* [30] also demonstrated the benefit of maintaining the coverage of data across different classes, which mitigates performance degradation at high pruning ratio. In our work, we build upon these observations and propose an importance sampling procedure which depends on the given pruning ratio and allows us to vary the importance of in addition to the difficulty of the samples retained.

Finally, there is a line of work aimed at online pruning, where a certain percentage of the data is pruned away continually at some specified interval (*e.g.*, every epoch, time elapsed, *etc.*) [2, 5, 6, 14]. These approaches are not directly comparable to ours since the pruning is applied continuously throughout the training process, rather than once at the start of training.

2.2 Data Distillation

Related to data pruning is the concept of data distillation. Rather than selecting a subset of images to retain, distillation aims to synthesize a set of representative samples to replace the original dataset. The idea was originally introduced in [27] and showcased promising results in distilling small image datasets (*e.g.*, MNIST, CIFAR-10) into just a single image per class. Subsequent works expanded on this by introducing label distillation, demonstrating the possibility of distilling datasets to less than one image per class [21]. Ongoing research has focused on refining both data and label distillation through enhanced optimization functions for learning the distilled data. A major limitation of data distillation methods is that they suffer from computational issues and do not scale well to synthesizing a larger number of images per class in order to be useful in practice. Sundar *et al.* [22] proposed to alleviate the computational burden by first pruning the original dataset and then performing data distillation on the remaining samples. For an in-depth review of data distillation, refer to the recent survey [18] and more recent methods such as [4].

3 Methods

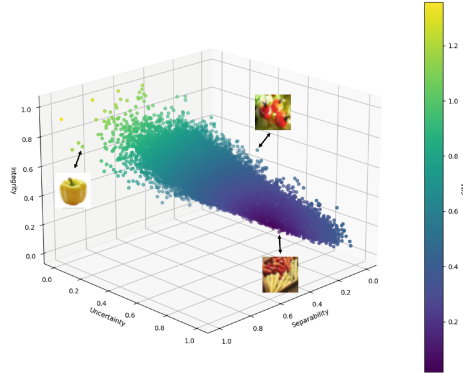


Fig. 1. Scatter plot of SIM scores for the CIFAR-100 dataset. The example images shown are from the “sweet pepper” class and are provided to give a visualization of the typical samples falling in those respective regions of the graph.

3.1 Problem Statement

Consider learning a classification model $f_\theta(\cdot)$ parameterized by θ using dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1, \dots, |\mathcal{D}|}$, which consists of a collection of samples x_i and corresponding class label $y_i \in \{1, \dots, C\}$ with C being the number of classes. Denote

$|\mathcal{D}|$ as the number of samples in the original dataset. We aim to develop a method that can identify a subset $\mathcal{D}' \subseteq \mathcal{D}$, such that $f_{\theta'}(\cdot)$ learned from \mathcal{D}' yields comparable accuracy with $f_{\theta}(\cdot)$. The reduction of data can be quantified through a pruning ratio defined as $\alpha = 1 - |\mathcal{D}'|/|\mathcal{D}|$. Given $0 \leq |\mathcal{D}'| \leq |\mathcal{D}|$, we have $\alpha \in [0, 1]$.

3.2 Data Separability

Data separability or discriminability describes the inherent and unchangeable properties of a given data sample due to noise and strength of the information portrayed [3]. Many previous pruning metrics measure the difficulty of each data sample as the pruning score. These scores are then used to rank the samples in order to determine which ones should be removed from the dataset. In the case of EL2N, the sample difficulty is measured as the l_2 norm of the error: $\|\hat{y}_i - y_i\|$. For Prototype scores, the difficulty is measured as the distance of each data sample to its mean class embedding: $\|f_{\theta}(x_i) - \frac{1}{n} \sum_{j=1}^n f_{\theta}(x_j)\|$, where n is the number of samples of corresponding class. However, neither of the measure considers the inter-class distance as an additional measure of sample difficulty. In case where all of the classes are well separated in the embedding space, this would not be an issue. However, when two or more classes are clustered together in the embedding space, then samples belonging to overlapping classes should be considered as more difficult than samples belonging to a perfectly separated class. For this reason, we choose to measure data separability using a modified version of the Recognizability Index (RI) [1]. Specifically, we define $d_{\theta}^P(x_i) = 1 - \cos(\phi_{y_i})$ as the positive distance of a sample x_i to its actual class center under model θ , where ϕ_{y_i} is the absolute angle between $f_{\theta}(x_i)$ and center embedding of class y_i . We then define $d_{\theta}^N(x_i) = 1 - \max_{j \in \{1, \dots, C\} \setminus \{y_i\}} \cos(\phi_j)$ as the negative distance of a sample to its nearest imposter class under model θ , where ϕ_j is the absolute angle between $f_{\theta}(x_i)$ and center embedding of class j . For both d_{θ}^P and d_{θ}^N , the center embedding is calculated as the average embeddings of class samples. Finally, data separability is computed as follows:

$$s_{\theta}(x_i) = \frac{d_{\theta}^N(x_i)}{d_{\theta}^P(x_i) + \epsilon} \quad (1)$$

where $\epsilon = 1e^{-7}$ is a small constant to avoid division by 0. Intuitively speaking, $s(x_i)$ reflects the difficulty to classify the data sample. An easily recognizable sample has small positive distance and large negative distance, thus a high value of $s(x_i)$. Otherwise $s(x_i)$ is low.

3.3 Data Integrity

A significant shortcoming of existing data pruning metrics is the failure to consider the integrity (*i.e.*, quality) of data samples. For a challenging sample, without capturing sample quality, the important distinction between recognizable and unrecognizable sample is missed. This oversight may lead to discarding

hard but useful samples (*e.g.*, profile-view of an object), instead of difficult samples due to low quality (*e.g.*, blurred image). To capture quality, we use the embedding norm as it has been previously shown to be correlated with sample quality [9] and is inexpensive to compute. In particular, given a classification model $f_\theta(\cdot)$, the embedding norm is defined as the l_2 -norm of the embedding vector.

$$e_\theta(x_i) = \|f_\theta(x_i)\|_2 \quad (2)$$

Empirically, it is observed that higher quality x_i often yields higher value of $e_\theta(x_i)$ and vice versa for a given model θ .

3.4 Model Uncertainty

So far, we have been focusing on deriving pruning metrics based on information possessed by the data itself. In practice, whether a model can utilize such information also impacts the effectiveness of pruned data. Therefore, we incorporate model uncertainty in our pruning metric, which is computed by Jensen-Shannon Divergence (JSD) of a set of expert models predictions $f(x_i)$ on a given data sample x_i . Similar approach was also adopted in prior work such as [12, 19]. As defined in Eq. (3), low uncertainty estimates are achieved when the entropy between individual model probabilistic distributions is low, meaning that all the models assign a similar probability distribution to different classes for given x_i . On the other hand, high entropy between probabilistic distributions on x_i indicates less agreement between the expert models on how to classify a given image example.

$$JSD(\{h_j\}_{j=1}^K | x_i) = H(M) - \frac{1}{K} \sum_{j=1}^K H(h_j(x_i)) \quad (3)$$

where $M = \frac{1}{K} \sum_{j=1}^K h_j(x_i)$ is the average probability distribution of x_i and K is the number of expert models. $H(h)$ is the Shannon entropy for distribution h . In practice, the expert models can be obtained by training different models using the entire dataset. h can be estimated as the softmax output of classification model. Intuitively speaking, more noisy and lower quality samples lead to higher uncertainty. Notice that $JSD(\cdot)$ is bounded between 0 and 1. For consistency with separability and integrity, we denote the compliment of the uncertainty as $c(x_i)$, such that higher value of $c(x_i)$ represents higher certainty and vice versa. $c(x_i)$ is computed as follows, which is also bounded between 0 and 1.

$$c(x_i) = 1 - JSD(\{h_j\}_{j=1}^K | x_i) \quad (4)$$

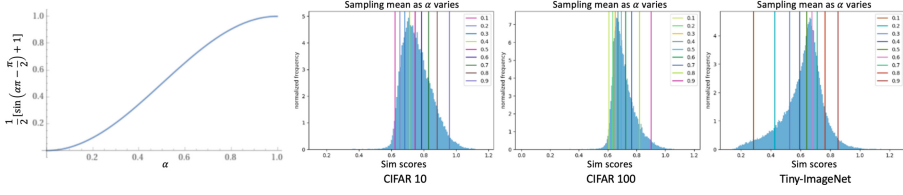


Fig. 2. Illustration of the sampling distribution mean at different pruning ratio α .

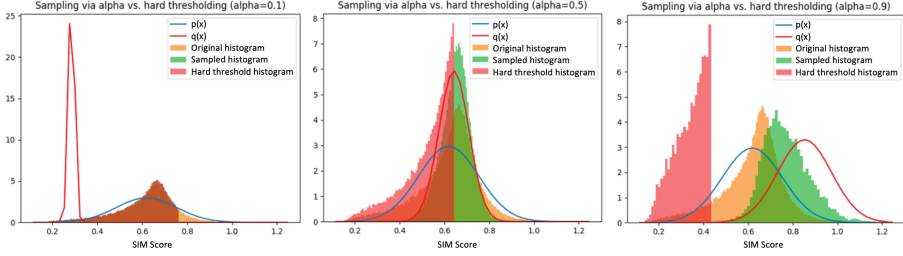


Fig. 3. Illustration of varying sampling distributions for Tiny-ImageNet dataset as pruning ratio α increases.

3.5 Derivation of SIM

We now discuss how to combine these three metrics together as the proposed SIM metric. First, we leverage the same set of K expert models used in uncertainty estimation to obtain aggregated separability and integrity defined as follows.

$$s(x_i) = \frac{1}{K} \sum_{j=1}^K s_{\theta_j}(x_i) \quad (5)$$

$$e(x_i) = \frac{1}{K} \sum_{j=1}^K e_{\theta_j}(x_i) \quad (6)$$

Then we normalize each metric $s(x_i)$, $e(x_i)$, and $c(x_i)$ separately by subtracting the minimum followed by dividing by the range so that each metric is spread between 0 and 1. Finally, we combine the three metrics in the following way:

$$g(x_i) = \sqrt{(1 - s(x_i))^2 + c(x_i)^2} - \sqrt{(1 - s(x_i))^2 + (1 - c(x_i))^2} \quad (7)$$

$$SIM(x_i) = \sqrt{g(x_i)^2 + e(x_i)^2} \quad (8)$$

The motivation for this formulation can be explained by the geometric interpretation of the two terms in Eq. (7). The first term, $\sqrt{(1 - s(x_i))^2 + c(x_i)^2}$ can be seen as the distance to the top-right corner of the normalized separability-uncertainty plane. This term is high for low data separability and low uncertainty samples. Samples falling in this region are assumed to be hard to recognize but potentially useful given that the uncertainty is low. The second term,

$\sqrt{(1 - s(x_i))^2 + (1 - c(x_i))^2}$ can be viewed as the distance to the bottom-right corner of the normalized separability-uncertainty plane. This term is penalizing samples with low separability and high uncertainty. Thus, the first term is trying to give a positive weighting towards retaining those difficult, but recognizable samples and the second term is giving a negative weighting towards low separability and high uncertainty samples. The distribution of SIM scores is illustrated for the CIFAR-100 dataset in Fig. 1. We also tried the formulation of $g'(x_i) = \sqrt{s(x_i)^2 + c(x_i)^2}$ but the performance is worse compared to $g(x_i)$.

An intuition as to why we combine each of these three metrics is shown in Fig. 6, which shows some example images from CIFAR-100 that have poor quality aspects that are captured by either $g(x_i)$, $e(x_i)$, or both. For example, according to the embedding norm $e(x_i)$, the images in subfigure (a) should have low scores due to the presence of blurry, low contrast images. Similarly, according to $g(x_i)$, the images in subfigure (b) should also have low scores due to the presence of multiple classes in the image. However, in all of these images, it seems reasonable to expect the classifier to learn to correctly identify the main objects. Indeed, if we had just used either $g(x_i)$ or $e(x_i)$ alone, then each of the images would have been likely pruned away early on from the training dataset. In contrast, by combining both $g(x_i)$ and $e(x_i)$, the resulting SIM scores actually fall towards the median and are retained for moderate values of α . Finally, we also retain the ability to reject cases of unrecognizable samples that exhibit degradations captured by both $g(x_i)$ and $e(x_i)$, like those in subfigure (c) of Fig. 6, which are likely not contributing positively towards the learning task.

3.6 SIM with Importance Sampling (SIMS)

Existing pruning methods usually sort samples by the pruning metric and retain the samples that meet a pre-defined threshold. Despite its simplicity, retaining the most difficult samples according to the pruning metric may lead to underfitting for large pruning ratios and datasets with more challenging classes. To address this limitation, we propose an adaptive sampling strategy based on the pruning ratio. It can be combined seamlessly with SIM and any other scalar-based pruning metric. We name our approach as SIMS, which consists of two key designs: i) varying the sample difficulty of pruned samples based on the pruning ratio and ii) sampling via a combination of class dependent and class independent distributions in order to avoid exacerbating bias in class-imbalanced data distributions.

Importance Sampling Based on Pruning Ratio. Our sampling strategy employs importance sampling [10] to emphasize the desired difficulty of samples during pruning. Rather than ranking the SIM scores from lowest to highest and retaining a percentage of the highest scoring samples based on the pruning ratio (α), the data to retain is selected via a sampling procedure. We construct a sampling distribution $q(x|\alpha)$, that is dependent on α , to assign an *importance* weight $w(x|\alpha)$ to each data sample x in the original data distribution $p(x)$ to

bias the sampling based on their *importance*. The *importance* is defined as the ratio between $q(x|\alpha)$ and $p(x)$ (*i.e.*, $w(x|\alpha) = \frac{q(x|\alpha)}{p(x)}$). Our intuition is that for small α , a large percentage of difficult samples can be retained because we have a lot of data samples remaining, whereas for large α , we have very few data samples retained so we need to retain the easy, representative examples.

In our implementation, both the original distribution $p(x)$ and sampling distribution $q(x|\alpha)$ assume the form of Normal distributions *i.e.*, $p(x) \sim \mathcal{N}(\mu_0, \sigma_0^2)$ and $q(x|\alpha) \sim \mathcal{N}(\mu, \sigma^2)$, where μ_0 and σ_0 are estimated from the training data directly. μ and σ are estimated based on α , μ_0 , and σ_0 . Specifically, μ is determined by Eq. (10), where F^{-1} is the inverse cumulative distribution function of $p(x)$ and t is the quantile function defined by Eq. (9). Notice t is a sinusoidal function of α , which increases monotonically with α . As illustrated in Fig. 2, smaller α reduces μ further so that $w(x|\alpha)$ is larger for x with smaller pruning score *i.e.* more difficult samples and vice versa.

$$t = \frac{1}{2} \left(\sin(\alpha\pi - \frac{\pi}{2}) + 1 \right) \quad (9)$$

$$\mu = F^{-1}(t; \mu_0, \sigma_0) \quad (10)$$

For σ , we simply use a scaled version of σ_0 as defined in Eq. (11). This is based off the observation that most of the high SIM scores for each class (see Fig. 2 in supplementary) contain similar, redundant samples. Therefore, as α increases, σ also increases so that we allow for extra diversity in the sampled images as the number of samples to retain becomes small.

$$\sigma = \alpha\sigma_0 \quad (11)$$

An illustration of the importance sampling procedure is given in Fig. 3 for Tiny-ImageNet [13] using SIM scores and sampling at various pruning ratios.

Combination of Class-Dependent and Class-Independent Sampling.

Our decision to include both class-dependent and class-independent sampling is motivated from the following two observations. First, we observed that the performance of pruning using the Prototype score can be improved by reserving a percentage of the pruned dataset for samples at the center of each cluster. In particular, CIFAR-10 has the best performance when 10% of the data is sampled from the class centers and 90% from the boundary (see Table 1 in the supplementary). Second, previous research has shown that data pruning via an overall score distribution for the entire dataset may exacerbate class imbalances [20]. Motivated by these observations, we adapt our importance sampling strategy to first sample a percentage of samples within each class (class-dependent sampling) and then sample the remaining percentage from the overall data distribution (class-independent sampling). Through an empirical analysis, the optimal performance was achieved by first sampling 5% of samples from each individual class and the remaining 95% from the overall data distribution.

4 Experimental Results

In this section, we provide a comparison of the proposed SIMS approach with various benchmark methods. We report the comparison considering both factors of classification accuracy and training time on the pruned datasets. We also explore the generalization performance of SIMS when models used for deriving pruning metric and performing classification are different. More qualitative analysis of pruning results is provided in the supplementary.

4.1 Datasets

We conducted our experiments using four image classification datasets. CIFAR-10 [11], CIFAR-100 [11], and Tiny-ImageNet [13] are well-known selections utilized in prior studies. CIFAR 10 and CIFAR 100 have 50,000 images with 10 and 100 classes, respectively, and Tiny-ImageNet has 100,000 images distributed equally among 200 classes. Finally, the large-scale, fine-grained dataset iNaturalist [26] has 675,170 images with a long-tailed distribution across 5,089 classes.

Table 1. Classification accuracy on CIFAR-10, CIFAR-100, Tiny-ImageNet, and iNaturalist datasets. Results are averaged across 3 repeated experiments. Best result in each column for each dataset is in **bold**.

[illegible]

4.2 Classification Accuracy

We assessed various baseline pruning metrics: Random, EL2N [16], Prototype [20], and Forgetting Scores [25] against the proposed SIMS on datasets mentioned in Sect. 4.1. We vary α from 0.1 to 0.9 with 0.1 increments for each experiment. For fair comparison across different methods, we used ResNet18 models for both calculating pruning metrics and classification for all methods. Furthermore, all experiments are conducted three times and the average results are reported. From the results in Table 1 and Table 2, we make the following observations.

First, SIMS outperforms the next best method, Forgetting Scores, on CIFAR-10, CIFAR-100, and iNaturalist datasets and is comparable on Tiny-ImageNet. Next, the biggest difference in performance is observed in the high α regime *i.e.*, $\alpha \in \{0.5, \dots, 0.9\}$, where SIMS outperforms Forgetting Scores across all datasets by a large margin. For example, the models trained on SIMS pruned datasets achieved, on average, a 22.1% increase in test classification accuracy over those trained on Forgetting Scores pruned datasets. Lastly, the incorporation of importance sampling substantially elevated the performance of both SIM (*i.e.* SIMS without importance sampling) and Forgetting Scores, increasing the average performance across the four datasets for $\alpha \in \{0.1, \dots, 0.9\}$ by 30.87% and 8.57% for SIM and Forgetting Scores, respectively. Thus, importance sampling is proved to be a useful strategy to pair with a general pruning metric in order to improve its performance.

Interesting to note is the high performance of random pruning averaged across all values of α for the iNaturalist dataset, although SIMS still performs significantly better for high pruning ratios ($\alpha \in \{0.5, \dots, 0.9\}$). We posit that given the relatively low performance and difficulty of this fine-grained classification dataset, the performance at low pruning ratios actually degrades rapidly when biasing the selection to retain low SIM score samples. This is because the classification problem is already difficult enough, that focusing on the boundary samples for each class (low SIM scores), rather than the representative samples (high SIM scores), is hurting the performance significantly. Thus, it seems that biasing the original focus to easier or more challenging samples depending on the initial full dataset performance may prove as a useful exploration for future work, especially on more challenging long-tailed and fine-grained datasets.

Table 2. Accuracy of Forgetting vs. Forgetting plus importance sampling across CIFAR-10, CIFAR-100, Tiny-ImageNet, and iNaturalist datasets, averaged across α from 0.1-0.9.

Metric	CIFAR-10	CIFAR-100	Tiny	iNaturalist
Forgetting	83.73	57.27	49.33	28.52
Forgetting + Sampling	87.82	61.23	47.12	36.04

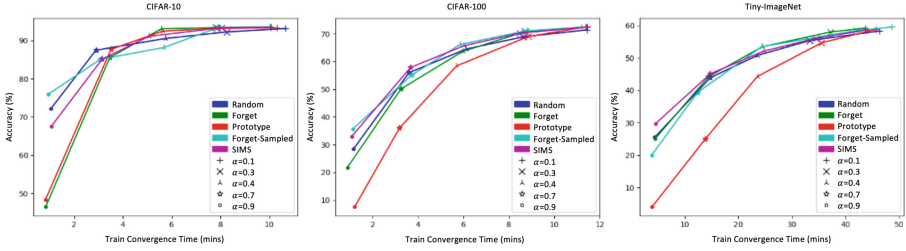


Fig. 4. Classification accuracy vs. training time of different pruning methods (Best viewed in color).

4.3 Training Time vs. Accuracy

One benefit of data pruning is reducing the training time compared to the full dataset, which is beneficial for tasks like hyper-parameter tuning and neural architecture search. Figure 4 presents a plot of training time vs. testing accuracy for different dataset using different pruning metrics at different pruning ratios. Comparing SIMS to the next best benchmark method, Forgetting Scores, we see a clear separation in terms of accuracy at higher pruning ratios (left half of the curve) with comparable training time. Where the efficiency advantage of different pruning metrics becomes evident is in the time required to compute the metrics, where we only train each expert model for 20 epochs for computing SIM scores, compared to Forgetting Scores which requires at least 75 epochs for each expert model [25]). Similarly, EL2N scores only require training for 20 epochs [16], but their accuracy is much lower across all of the datasets used in our evaluations.

Table 3. Accuracy of all possible combinations of metrics and sampling procedures. The best result in each column is in **bold**.

Methods Ablation	CIFAR-10	CIFAR-100	Tiny	Avg.
S	87.53	60.12	45.40	64.35
I	86.57	60.33	46.22	64.37
M	88.04	58.83	46.51	64.46
S + I	86.40	56.93	43.53	62.29
S + M	87.83	60.44	46.21	64.83
I + M	86.30	60.41	42.55	63.09
S + I + M (proposed)	87.56	61.39	49.20	66.05
No sampling	84.76	52.55	39.92	59.08
0% class sampling	87.82	61.23	47.12	65.39
1% class sampling	87.31	61.20	49.21	65.91
5% class sampling (proposed)	87.56	61.39	49.20	66.05
100% class sampling	87.26	61.02	49.22	65.83

Table 4. Cross-model generalization. Each column represents different classification models and datasets. Each row represents different pruning metric and pruning model is ResNet18.

Pruning metric	ResNet18		ResNet50	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Forgetting Scores	87.82	61.23	91.72	73.76
SIMS	87.31	61.20	92.09	74.17

4.4 Ablation Analysis

An in-depth ablation study was conducted on the individual components of SIMS. For all studies in this section, we used $K = 10$ expert ResNet18 models for pruning and used a different ResNet18 model trained on the pruned datasets for classification. Same as Sect. 4.2, we vary α from 0.1 to 0.9 with 0.1 increments and report average accuracy on different datasets. First, we ablated on the various combinations of the three metric components of SIM scores, including separability (S), integrity (I), and model uncertainty (M). The results presented in Table 3 upper subsection show that the best average performance of 66.05% is obtained in the last row from the combination of all three metrics *i.e.* SIM. We also notice the best performing individual metric is model uncertainty. And combining two metrics does not necessarily improve overall performance *e.g.* S+I perform worse than S or I used separately.

The second ablation study was performed on the various components of the importance sampling procedure employed by SIMS. The results are given in Table 3 lower subsection. First, we observe a drastic decrease in performance without sampling (row 1) compared to any one of remaining four sampling procedures (row 2–5). Next, we analyze the effect of different variants of sampling by changing the within-class ratio, where 0% means performing sampling on entire dataset, and 100% means each class is equally pruned. From rows 2–5, we can see that as the difficulty and number of classes in the dataset increases, the benefit of sampling a percentage of the data within classes becomes more significant. In particular, we see about 2.1% decrease in performance on Tiny-ImageNet when sampling strictly from the overall distribution compared to first sampling within each class and then sampling the remaining amount from the overall distribution (row 2 vs. row 3–5). The best overall performance across the three datasets is obtained with an initial 5% sampling within class and remaining 95% sampling from the overall distribution. To gain further insight into the benefit of importance sampling, we plotted pruned vs. retained samples on CIFAR-10 in the t-SNE space in Fig. 5. Notice that importance sampling results in better coverage of each class distribution. Additional results are in supplementary Fig. 1.

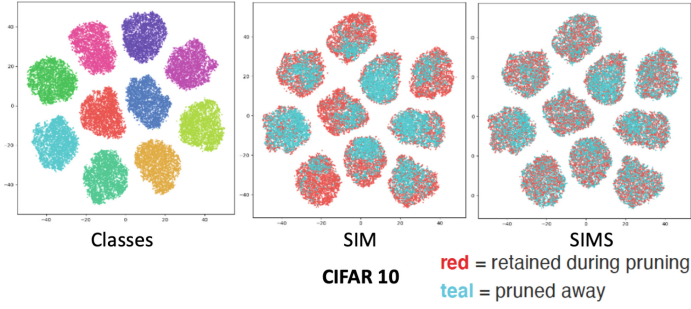


Fig. 5. T-SNE visualization for CIFAR-10 comparing SIM to SIMS.

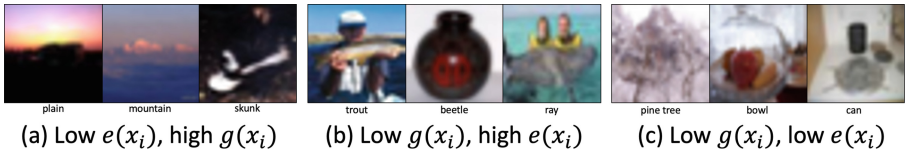


Fig. 6. Example images from CIFAR-100 with poor qualities such as blurriness, low contrast, and multiple classes. The advantage of computing SIM scores is evident in the different quality aspects captured by $e(x_i)$ and $g(x_i)$, where many difficult but useful samples would be rejected if either metric was used alone.

4.5 Cross-Model Generalization

Finally, we conducted generalization experiments to test whether SIMS performs well when training downstream classification models on pruned datasets obtained via a different model architecture. From Table 4, SIMS performs better than Forgetting Scores with a ResNet50 model trained on the pruned datasets derived with a ResNet18 model. This suggests that some of the upfront costs of computing SIMS scores can be reduced by training expert models with a smaller architecture and subsequently training larger, downstream models on the pruned datasets.

5 Conclusion

We proposed a data pruning approach (*a.k.a.* SIMS) which improves upon state-of-the-art pruning methods in terms of scalability and generalizability, as demonstrated on various datasets including the large-scale and long-tailed distribution dataset, iNaturalist. We showed that our method performs especially well at high pruning ratios and challenging fine-grained dataset compared to state-of-the-art. We showed that a key component of our proposed pruning procedure, importance sampling, was not only instrumental in improving the performance of the proposed SIM metric, but also can boost performance of other pruning metrics *e.g.*, Forgetting Scores. Furthermore, we showed that SIMS can generalize better

across unseen model architectures, which is desirable from practical perspective. Our future direction is reducing the cost of training expert models required for achieving best results, integrating the full dataset performance as a prior in the importance sampling, and extending the metric to unsupervised learning of pruning models. Furthermore, even though we focused on the classification task in this paper, the idea of using SIM to characterize the data separability, data integrity, and model uncertainty is generic. The importance sampling process can also be extended to other tasks, such as detection and segmentation, since it is based on model uncertainty. Therefore, we believe the proposed SIMS is applicable to other tasks.

References

1. Chai, J.C.L., Ng, T.S., Low, C.Y., Park, J., Teoh, A.B.J.: Recognizability embedding enhancement for very low-resolution face recognition and quality estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9957–9967 (2023)
2. Chitta, K., Álvarez, J.M., Haussmann, E., Farabet, C.: Training data subset search with ensemble active learning. *IEEE Trans. Intell. Transp. Syst.* **23**(9), 14741–14752 (2021)
3. Duda, R.O., Hart, P.E., et al.: *Pattern Classification*. Wiley (2006)
4. He, Y., Xiao, L., Zhou, J.T., Tsang, I.: Multisize dataset condensation. In: *The Twelfth International Conference on Learning Representations* (2024)
5. Khan, M.A., Hamila, R., Menouar, H.: CLIP: train faster with less data. In: *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 34–39. IEEE (2023)
6. Killamsetty, K., Evfimievski, A.V., Pedapati, T., Kate, K., Popa, L., Iyer, R.: Milo: model-agnostic subset selection framework for efficient model training and tuning. *arXiv preprint [arXiv:2301.13287](https://arxiv.org/abs/2301.13287)* (2023)
7. Killamsetty, K., et al.: Automata: gradient based data subset selection for compute-efficient hyper-parameter tuning. *Adv. Neural. Inf. Process. Syst.* **35**, 28721–28733 (2022)
8. Killamsetty, K., Durga, S., Ramakrishnan, G., De, A., Iyer, R.: Grad-match: gradient matching based data subset selection for efficient deep model training. In: *International Conference on Machine Learning*, pp. 5464–5474. PMLR (2021)
9. Kim, M., Jain, A.K., Liu, X.: AdaFace: quality adaptive margin for face recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18750–18759 (2022)
10. Kloek, T., Van Dijk, H.K.: Bayesian estimates of equation system parameters: an application of integration by monte carlo. *Econometrica: J. Econ. Soc.* **46**, 1–19 (1978)
11. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
12. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: *Advances in Neural Information Processing Systems*, vol.30 (2017)
13. Le, Y., Yang, X.: Tiny ImageNet visual recognition challenge. *CS 231N* **7**(7), 3 (2015)

14. Li, J., He, L., Ren, S., Mao, R.: Data fine-pruning: a simple way to accelerate neural network training. In: Zhang, F., Zhai, J., Snir, M., Jin, H., Kasahara, H., Valero, M. (eds.) NPC 2018. LNCS, vol. 11276, pp. 114–125. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-05677-3_10
15. Park, D., Choi, S., Kim, D., Song, H., Lee, J.G.: Robust data pruning under label noise via maximizing re-labeling accuracy. In: Thirty-Seventh Conference on Neural Information Processing Systems (2023)
16. Paul, M., Ganguli, S., Dziugaite, G.K.: Deep learning on a data diet: finding important examples early in training. *Adv. Neural. Inf. Process. Syst.* **34**, 20596–20607 (2021)
17. Pote, T., Adnan, M., Yargic, Y., Ioannou, Y.: Classification bias on a data diet. In: Conference on Parsimony and Learning (Recent Spotlight Track) (2023)
18. Sachdeva, N., McAuley, J.: Data distillation: a survey. *arXiv preprint [arXiv:2301.04272](https://arxiv.org/abs/2301.04272)* (2023)
19. Shoshan, A., et al.: Asymmetric image retrieval with cross model compatible ensembles. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1–11 (2024)
20. Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., Morcos, A.: Beyond neural scaling laws: beating power law scaling via data pruning. *Adv. Neural. Inf. Process. Syst.* **35**, 19523–19536 (2022)
21. Sucholutsky, I., Schonlau, M.: Less than one-shot learning: learning n classes from $m \ll n$ samples. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 9739–9746 (2021)
22. Sundar, A.S., Keskin, G., Chandak, C., Chen, I.F., Ghahremani, P., Ghosh, S.: Prune then distill: dataset distillation with importance sampling. In: ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023)
23. Susan, S., Kumar, A.: The balancing trick: optimized sampling of imbalanced datasets-a brief survey of the recent state of the art. *Eng. Rep.* **3**(4), e12298 (2021)
24. Tan, H., et al.: Data pruning via moving-one-sample-out. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
25. Toneva, M., Sordoni, A., des Combes, R.T., Trischler, A., Bengio, Y., Gordon, G.J.: An empirical study of example forgetting during deep neural network learning. In: International Conference on Learning Representations (2019)
26. Van Horn, et al.: The INaturalist species classification and detection dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8769–8778 (2018)
27. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. *arXiv preprint [arXiv:1811.10959](https://arxiv.org/abs/1811.10959)* (2018)
28. Xia, X., Liu, J., Yu, J., Shen, X., Han, B., Liu, T.: Moderate coreset: a universal method of data selection for real-world data-efficient deep learning. In: The Eleventh International Conference on Learning Representations (2022)
29. Yang, S., Xie, Z., Peng, H., Xu, M., Sun, M., Li, P.: Dataset pruning: reducing training data by examining generalization influence. *arXiv preprint [arXiv:2205.09329](https://arxiv.org/abs/2205.09329)* (2022)
30. Zheng, H., Liu, R., Lai, F., Prakash, A.: Coverage-centric coreset selection for high pruning rates. In: The Eleventh International Conference on Learning Representations (2022)



Label-Specific Multi-label Classification with Entropy Guided Clustering

Jiaxuan Li[✉], Tong Zhu[✉], Xiaoyan Zhu[✉], and Jiayin Wang[✉]

Xi'an Jiaotong University, Xi'an, China

{lijiaxuan, sae11011}@stu.xjtu.edu.cn, zhu.xy@xjtu.edu.cn,
wangjiayin@mail.xjtu.edu.cn

Abstract. Multi-label classification deals with the problem where each instance is associated with multiple labels. To discriminate the label difference, each label can be modeled in its specific feature subset derived from the original feature space. In these label-specific methods, the mainstream is to generate new features by analyzing the distance relationship between data points and the clusters they aggregate into. However, it is difficult to determine how many clusters are required, and clustering algorithms are often unstable. In this paper, we take entropy to measure clustering quality and establish a novel model to quantitatively determine the number of clusters. Besides, a novel conception of entropy similarity is proposed to pairwise measure label correlation and enable clustering ensemble to improve model robustness. Experiments on 12 benchmark datasets validate the effectiveness of the proposed method.

Keywords: Multi-label classification · Label-specific feature · Entropy · Clustering

1 Introduction

In traditional supervised learning, each instance is associated with one label. Differently, multi-label classification deals with the problem that each instance can be associated with a set of labels. For example, in scene classification, an image consisting of 1080*1920 pixels can contain many labels simultaneously, such as buildings, plants, animals, etc. Nowadays, multi-label classification has been applied in many fields, such as scene classification [2, 32], text categorization [20, 24] and bioinformatics [23, 33].

Over the past decades, lots of algorithms have been proposed to address multi-label classification tasks. Typically, problem transformation methods [1, 15, 19] convert the multi-label problem to single-label ones, and then binary classification algorithms can be employed, while algorithm adaptation methods [3, 5, 30] directly expend single-label algorithms to their multi-label versions.

In all the aforementioned methods, multi-label model is constructed on the feature space and then employed to make predictions for all labels. However, a feature may only be closely associated with a part of labels. For the scene

classification example, features related to green pixels tend to be more useful in predicting whether a picture contains plants, while blue pixels for sky or sea. Accordingly, some researchers propose the concept of label-specific features [10, 29], that is, for each label, only the most pertinent and discriminative features should be taken for model induction.

LIFT [29] firstly proposes to generate label-specific features for multi-label classification. It employs clustering analysis to obtain positive and negative clusters respectively for every label class, and the distances between all instances and cluster centers are formed as new features. LIFT effectively extracts different features for different labels, but it still has some drawbacks, such as unstable clustering and the deficiency of label correlation. Next, researchers extend LIFT from various aspects, e.g. taking spectral clustering instead of k-means [28], guiding clustering by entropy [27] or spatial topological information [7], incorporating deep learning approach [9], enhancing model stability by base learner ensemble [22] or clustering ensemble [21, 25].

Overall, above improvements fail to determine how many clusters are wanted given a multi-label dataset, which heavily limits the effectiveness of clustering analysis. To cope with this issue, a novel method named ECL (Entropy guided Clustering for Label-specific learning) is proposed in this paper, using entropy to measure the uncertainty degree of clusters, and on the top of this definition, an objective function is established to quantitatively determine the number of clusters. Besides, for the issue of unstable clustering, clustering ensemble is valid to improve the performance of a cluster with other clusters [25]. In this paper, a conception of entropy similarity is proposed to guide clustering ensemble and improve the quality of label-specific features by considering label correlation.

To summarize, ECL systematically improves the clustering-based label-specific features generation method through the definition and application of entropy in multi-label classification. The contributions of ECL can be highlighted as follows:

- A novel paradigm of multi-label entropy is presented, considering the distribution of both labels and instances.
- An objective function is established to provide a concrete number of clusters for distinct label, guaranteeing the clustering quality.
- A novel entropy-based label similarity is designed, which effectively guides clustering ensemble to improve clustering robustness and label correlation exploration.

2 Preliminaries

To benefit the analysis on related work and the introduction to the proposed method, the steps of clustering-based label-specific features generation [29] will be introduced. Formally, the main notations are listed in Table 1.

LIFT [29] extracts new features from the distribution of training instances. Firstly, label independence methodology is taken to transform the original multi-label dataset $\mathcal{D} = \{\mathcal{X}, Y_1, \dots, Y_q\}$ to q individual single-label datasets $\{\mathcal{D}_j =$

Table 1. Main Notations

Notation	Description
$\mathcal{X} = \{X_1, X_2, \dots, X_d\} \in \mathbb{R}^d$	The d -dimensional feature space
$\mathcal{Y} = \{Y_1, Y_2, \dots, Y_q\} \in \{0, 1\}^q$	The label space with q labels
$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid 1 \leq i \leq n\}$	The training dataset
$\mathbf{x}_i, \mathbf{y}_i = \{y_{i1}, \dots, y_{iq}\}$	The i -th instance and its corresponding label vector
$Y_j, \lambda_j = \{y_{1j}, \dots, y_{nj}\}$	The j -th label and its corresponding label column

$(\mathcal{X}, Y_j) \mid 1 \leq j \leq q\}$. Since the label space is disjointly partitioned and label-specific features are independently generated, only Y_j is taken as an example in the following.

Secondly, \mathcal{D}_j is further divided to positive and negative subsets based on the label value:

$$\begin{aligned}\mathcal{P}_j &= \{\mathbf{x}_i \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}, y_{ij} = 1\} \\ \mathcal{N}_j &= \{\mathbf{x}_i \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}, y_{ij} = 0\}\end{aligned}\quad (1)$$

Thirdly, \mathcal{P}_j is partitioned into several disjoint clusters with k-means clustering algorithm and so do \mathcal{N}_j . $\mathcal{T}_j = \{\mathbf{p}_j^1, \mathbf{p}_j^2, \dots, \mathbf{p}_j^{k_j}; \mathbf{n}_j^1, \mathbf{n}_j^2, \dots, \mathbf{n}_j^{k_j}\}$ are the cluster centers with the number of clusters is:

$$k_j = r_{atio} \cdot \min \{|\mathcal{P}_j|, |\mathcal{N}_j|\} \quad (2)$$

where r_{atio} is a predefined parameter.

Lastly, the distances between instances in \mathcal{D}_j and $2k_j$ cluster centers are seen as new features specific to Y_j . Here, a mapping $\phi_j : \mathcal{X} \rightarrow \mathcal{Z}_j$ from the original d -dimensional feature space to newly generated $2k_j$ -dimensional feature space is constructed as follow:

$$\phi_j(\mathbf{x}) = \left\{ d(\mathbf{x}, \mathbf{p}_j^1), \dots, d(\mathbf{x}, \mathbf{p}_j^{k_j}), d(\mathbf{x}, \mathbf{n}_j^1), \dots, d(\mathbf{x}, \mathbf{n}_j^{k_j}) \right\} \quad (3)$$

where $d(\cdot, \cdot)$ represents the Euclidean distance between two instances.

3 Related Work

There are two categories of methods to find discriminate features for distinct label from the original feature set, namely selection and generation.

The selection methods recommend the most suitable features specific to distinct label. LLSF [10] first proposes this paradigm and build a multi-label regression model with ℓ_1 -norm regularizer. As the ℓ_1 -norm regularizer makes the weights for features sparse, each label obtains its own distinct feature subset. Then, many strategies are presented to enhance the label correlation learning ability, e.g. updating the prediction outputs according to the label interrelations [11], applying instance similarity consideration into the objective function [8],

and employing manifold regularization [14] in terms of label-wise and instance-wise at the same time [13]. However, these regression-based methods in practical applications rarely provide the specific features for each label, as the ℓ_1 -norm regularizer cannot ensure an absolute sparsity under the realistic optimization methods. More commonly, some features are assigned with larger weights, and instead, some features obtain smaller weights.

LIFT [29] is the first method of generating label-specific features. It independently treats each label, and for each label class, the instances are first partitioned into k clusters with k -means clustering. Subsequently, distances between cluster centers and instances are viewed as label-specific features, with the assumption that a cluster center can represent the instances in this cluster. The defects of LIFT derive from the instability of k -means clustering, i.e., 1) how large should k be set, and 2) how to avoid its sensitivity to initial cluster center setting. For the first issue, ML-LEC [27] takes label entropy to measure the uncertainty of training set, and creates more clusters for label classes with larger uncertainty. Some scholars have also proposed improvements in other ways, e.g. guiding clustering process with spatial topological information [7], and replacing k -means clustering with other clustering approach [28]. However, the existing methods can only qualitatively analyze the shortcomings of clustering and do some modifications, but fail to provide a number of clusters quantitatively.

For the second issue, LIFTACE [25] enable clustering ensemble with label similarity to eliminate the randomness in each clustering process. Besides, ELIFT [22] takes a weighted ensemble of LIFT to improve model robustness, LETTER [6] reconstructs two subsets of new label-specific features on the top of instance level and feature level, respectively, LF-LELC [26] further builds the label correlation learning mechanism after label-specific features generation. Nonetheless, the aforementioned methods generally consider label similarity, e.g. cosine similarities between label columns, to learn label correlation and enhance model robustness. While in the real-world multi-label classification, the label space is sparse, resulting in the difficulty in label similarity measurement.

4 The Proposed Method

In previous clustering-based label-specific features methods, two issues have not been fully addressed: 1) the number of clusters; 2) the stability of clustering. In this section, the solution to these two issues by ECL will be described in Sect. 4.1 and Sect. 4.2, respectively. In Sect. 4.3, the whole process of model induction is summarized.

4.1 Determining the Number of Clusters with Entropy

Cluster Entropy. Before discussing how many clusters are wanted, we first define the conception of cluster entropy in this subsection.

In information theory, entropy is taken to measure the uncertainty degree of a variable. For example, the entropy of label Y_j is:

$$E(Y_j) = - \sum_{y \in \{0,1\}} P(Y_j = y) \log P(Y_j = y) \quad (4)$$

Furthermore, considering that feature X_j may contain m_j , $m_j \geq 2$ values, we define the frequency of its i -th value is P_{ij} , and the entropy of this feature is:

$$E(X_j) = - \sum_{i=1}^{m_j} P_{ij} \log P_{ij} \quad (5)$$

where m_j will be a hyper-parameter if X_j is continuous-valued. In this paper, we discretize continuous variables using the manner of quantiles, that is, for each continuous-valued feature, its values are divided into m_j folds according to value ranking and each part has n/m_j values.

For a multi-label dataset \mathcal{D} containing d features and q labels, we can define its entropy as:

$$E(\mathcal{D}) = \frac{1}{d+q} \left(\sum_{j=1}^d E(X_j) + \sum_{j=1}^q E(Y_j) \right) \quad (6)$$

In clustering-based label-specific features generation, the labels are independently treated and the clustering analysis is performed in the single-label formed datasets $\mathcal{P}_j, \mathcal{N}_j$. To balance positive and negative classes, we set the same number of clusters for positive and negative instances, and perform clustering analysis in the instances with majority class:

$$\mathcal{M}_j = \begin{cases} \mathcal{P}_j, & \text{if } |\mathcal{P}_j| > |\mathcal{N}_j| \\ \mathcal{N}_j, & \text{if } |\mathcal{P}_j| \leq |\mathcal{N}_j| \end{cases} \quad (7)$$

where $|\cdot|$ denotes the number of instances in a dataset.

Suppose that \mathcal{C} is a cluster in \mathcal{M}_j , \mathcal{C} is essentially a subset of \mathcal{M}_j , and its corresponding cluster entropy can be formulated as:

$$E(\mathcal{C}) = \frac{1}{d+q-1} \left(\sum_{i=1}^d E(X_i) + \sum_{i=1, i \neq j}^q E(Y_i) \right) \quad (8)$$

where $E(X_i)$ and $E(Y_i)$ are the entropies of the i -th feature and i -th label in terms of the instances in \mathcal{C} .

Objective Function. Theoretically, in clustering-based label-specific methods, each cluster center is regarded as the representative instance of all instances in this cluster. To ensure the representative performance, two principles related to the entropy of each partitioned cluster should be observed:

- Principle 1: The entropy of each cluster should not be too large. Otherwise, each cluster would contain too many instances to make a cluster center to represent them. Such risk is shown as Fig. 1(a), where clusters corresponding to different classes are overlapped and a cluster center will be unable to represent the instances in the cluster.
- Principle 2: The entropy of each cluster should not be too small. If each cluster center represents only a few instances, it would be meaningless to extract features via clustering, and such phenomenon can be seen in Fig. 1(b). Besides, as the number of label-specific features is proportional to the number of clusters, too many clusters denote a complex label-specific feature space in high-dimension. Even, in the extreme case, if each instance is considered as a cluster, instances between any instance pairs would be viewed as label-specific features.

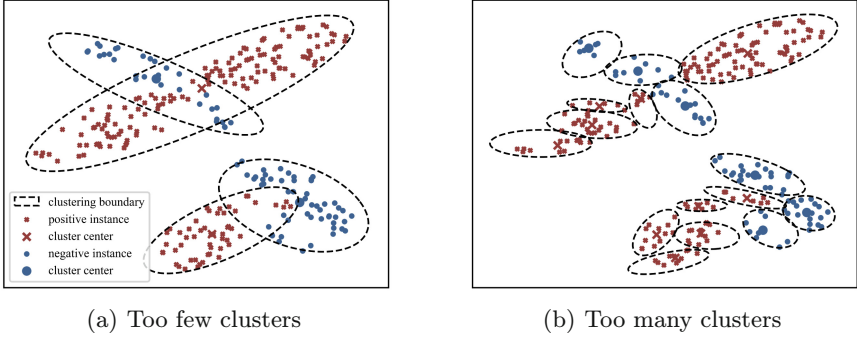


Fig. 1. An example of the number of clusters.

To cover these two principles, ECL establishes an objective function for each label class to obtain an optimal relation between the number of clusters and the entropy of partitioned clusters.

Firstly, the Principle 1 can be realized by minimizing the cluster entropy of each cluster. So, the objective function can be established to minimize the weighted sum of the entropies of all clusters:

$$\min_{k_j} \sum_{i=1}^{k_j} \frac{|\mathcal{C}_{ij}|}{|\mathcal{M}_j|} E(\mathcal{C}_{ij}) \quad (9)$$

where k_j is the number of clusters, and \mathcal{C}_{ij} is the i -th cluster in \mathcal{M}_j .

Secondly, following the Principle 2, the number of clusters should be as small as possible. Such requirement can be summarized as a ℓ_1 -norm regularization and the objective function can be defined as:

$$\min_{k_j} \sum_{i=1}^{k_j} \frac{|\mathcal{C}_{ij}|}{|\mathcal{M}_j|} E(\mathcal{C}_{ij}) + \beta \cdot k_j \quad (10)$$

where β is a trade-off parameter.

Optimization. In entropy calculation, a fundamental assumption is that the fewer clusters, the lower the entropy of each of them. As the number of clusters increasing, instances in same cluster will located in a narrow local. So, the first item in Eq. 10 are approximately monotonic decreasing and the second item will undoubtedly increase. Eventually, a conclusion can be obtained the objective function is decreasing first and then increasing.

To cope with the integer optimization problem, ECL takes the trisection method to find the minimum value of the objective function $f(k_j)$ in Eq. 10. Firstly, an initial search region $R = (a, b)$ should be predefined. Secondly, the function value in the two trisection points $c_1 = (2a+b)/3, c_2 = (a+2b)/3$ should be calculated. Thirdly, repeatedly update the search region:

$$R = \begin{cases} (a, c_2), & \text{if } f(c_1) \leq f(c_2) \\ (c_1, b), & \text{if } f(c_1) > f(c_2) \end{cases} \quad (11)$$

Repeat the last two steps, until get the minimum point.

ECL initiates different search intervals for different labels. For Y_j , the initial search interval is set as $[1, \min \{|\mathcal{P}_j|, |\mathcal{N}_j|\}]$, as the number of clusters should be less than the number of instances.

4.2 Incorporating Clustering Ensemble with Entropy-Based Label Similarity

Inspired by LIFTACE [25], we take clustering ensemble to enhance model stability and label correlation learning ability, where the clustering result of each label can be updated due to those of other labels. The core idea is that the more related two labels are, the more similar the clustering results corresponding to them. To represent clustering result, let $I(\mathbf{x}_i)$ denote the index of cluster where \mathbf{x}_i is located, so the instance similarity matrix of Y_j can be written as $W_j \in \{0, 1\}^{n \times n}$:

$$[W_j]_{ab} = \begin{cases} 1, & \text{if } I(\mathbf{x}_a) = I(\mathbf{x}_b) \\ 0, & \text{if } I(\mathbf{x}_a) \neq I(\mathbf{x}_b) \end{cases} \quad (12)$$

Then, W_j can be updated according to $\{W_i \mid i = 1, \dots, q, i \neq j\}$:

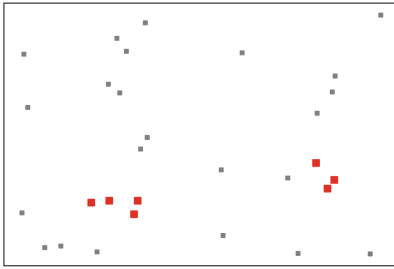
$$W'_j = \frac{\sum_{i=1}^q S_{ij} W_i}{\sum_{i=1}^q S_{ij}} \quad (13)$$

where S_{ij} is the similarity between i -th label and j -th label. Such pairwise similarity can be measured in various ways, e.g. Cosine, Jaccard, Euclidean, and

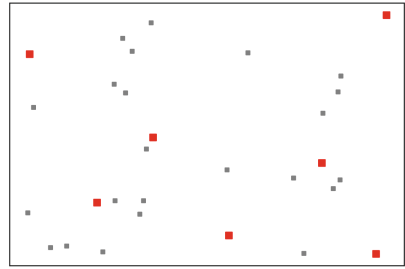
Hamming. Among them, Cosine similarity is one of the most popular in multi-label classification [25].

However, these measures only consider the difference in the label space, but neglects the distribution of instances corresponding to the label. Therefore, we design a new entropy-based label similarity measure. Specifically, for labels Y_i and Y_j , corresponding instances can be divided into four subsets, i.e., $\mathcal{D}_{11} \leftarrow Y_i \cap Y_j$, $\mathcal{D}_{01} \leftarrow \bar{Y}_i \cap Y_j$, $\mathcal{D}_{10} \leftarrow Y_i \cap \bar{Y}_j$, $\mathcal{D}_{00} \leftarrow \bar{Y}_i \cap \bar{Y}_j$ where Y_i represents the set of instances with label Y_i , while \bar{Y}_i denotes the opposite.

Therefore, the $\mathcal{D}_{01}, \mathcal{D}_{10}$ contain the instances that have different label values for Y_i, Y_j , and we call these instances as difference instances. Apparently, the distribution of difference instances implies the dissimilarity between labels. For example, the instances of $\mathcal{D}_{01}, \mathcal{D}_{10}$ in narrow region can be caused by imbalanced data distribution, while the dissimilarity cannot be ignored if these instances are scattered. Such problem can be shown in Fig. 2, where gray pixels are instances of $\mathcal{D}_{11}, \mathcal{D}_{00}$ and red big pixels come from $\mathcal{D}_{01}, \mathcal{D}_{10}$. Both two distributions in Fig. 2(a) and Fig. 2(b) have 7 difference instances, but the latter one has greater distribution difference than the former one.



(a) Difference instances in concentrated distribution



(b) Difference instances in scattered distribution

Fig. 2. Different distributions of instances lead to different degrees of similarity.

To address this issue, a novel entropy-based label similarity measure is defined as:

$$S_{ij} = 1 - \frac{|\mathcal{D}_{01}| E(\mathcal{D}_{01}) + |\mathcal{D}_{10}| E(\mathcal{D}_{10})}{\sum_{l \in \{00, 01, 10, 11\}} |\mathcal{D}_l| E(\mathcal{D}_l)} \quad (14)$$

Lastly, the clustering results can be updated with Eq. 13, following the graph-based clustering ensemble methods [18]. In this paper, SPEC algorithm [17] is employed due to its effectiveness and simplicity.

4.3 Model Induction

In this subsection, we will introduce all the steps of ECL, from label-specific features generation to model construction and unknown instances inference. The whole process of ECL is shown in Algorithm 1.

Algorithm 1. The framework of ECL

Input: Multi-label training dataset \mathcal{D} , Hyperparameter: β, m_j .

Output: The prediction \mathbf{y}_{new} given an unknown instance \mathbf{x}_{new} .

I Generating label-specific features.

- 1: **for** j in q **do**
- 2: $\mathcal{N}_j, \mathcal{P}_j \leftarrow \mathcal{D}, j$: partition \mathcal{D} into two parts with respect to Y_j with Eq. 1
- 3: $\mathcal{M}_j \leftarrow \mathcal{N}_j, \mathcal{P}_j$: get the majority class instances of \mathcal{D}_j with Eq. 7
- 4: $k_j \leftarrow \mathcal{M}_j$: optimize to get the number of clusters with Eq. 10.
- 5: $\mathcal{T}_j = \{\mathbf{n}_j^1, \dots, \mathbf{n}_j^{k_j}, \mathbf{p}_j^1, \dots, \mathbf{p}_j^{k_j}\} \leftarrow \mathcal{N}_j, \mathcal{P}_j, k_j$: obtain $2k_j$ cluster centers.
- 6: **end for**
- 7: $S \leftarrow \mathcal{D}$: calculate the label similarity matrix with Eq. 14.
- 8: **for** j in q **do**
- 9: $W_j \leftarrow \mathcal{T}_j, \mathcal{P}_j, \mathcal{N}_j$: calculate instance similarity matrix with Eq. 12.
- 10: **end for**
- 11: **for** j in q **do**
- 12: $W'_j \leftarrow W, S$: update the instance similarity matrix with Eq. 13
- 13: $\mathcal{T}'_j \leftarrow W'_j$: update the cluster centers with SPEC algorithm
- 14: **end for**
- 15: **for** j in q **do**
- 16: $\phi_j \leftarrow \mathcal{D}, \mathcal{T}'_j$: generate a mapping for label-specific features with Eq. 3.
- 17: **end for**

II Training the model.

- 18: **for** j in q **do**
- 19: $\mathcal{L}_j \leftarrow (\phi_j, Y_j)$: induce the j -th single-label learner.
- 20: **end for**
- 21: $\mathcal{L} = \{\mathcal{L}_j\}$: the multi-label learner.

III Predicting unknown instance.

- 22: $\mathbf{x}'_{new} \leftarrow \mathbf{x}_{new}, \Phi$: convert the features of \mathbf{x}_{new} to label-specific features.
 - 23: $\mathbf{y}_{new} \leftarrow \mathcal{L}, \mathbf{x}'_{new}$: make prediction For \mathbf{x}_{new} .
-

5 Experimental Study

5.1 Experimental Setup

Benchmark Datasets. In the experiments, 12 datasets from 7 different domains are considered. All these datasets can be downloaded from the website of KDIS¹. Details about the used datasets are shown in Table 2, including

¹ All datasets are available at <http://www.uco.es/kdis/mlresources/>.

names and domains of the datasets, number of instances (n), features (d) and labels (q) of the datasets.

Table 2. Characteristics of the Experimental Data Sets.

DataID	Dataset	Domain	n	d	q
#1	Guardian	Text	302	1000	6
#2	CHD_49	Medicine	555	49	6
#3	Emotions	Music	593	72	6
#4	Foodtruck	Recommend	407	21	12
#5	Image	Image	2000	294	5
#6	Scene	Image	2407	294	6
#7	WaterQuality	Chemistry	1060	16	14
#8	Yeast	Biology	2417	103	14
#9	EukaryotePA	Biology	7766	440	22
#10	HumanPA	Biology	3106	440	14
#11	CAL500	Music	502	68	174
#12	Langlog	Text	1460	1004	75

In the experiments, the original dataset is divided into training and testing datasets using Iterative Stratified method [16] in the partition of 67% and 33%, respectively. To obtain more reliable results, all algorithms are repeated 5 times and then averaged.

Comparison Methods. To report reliable comparison results, ECL is compared with 7 baseline algorithms. The 7 comparison methods can be divided into 3 categories, according to their relationship with label-specific (LS) approach,

- Non-LS methods: LRDG [31] and SRLG [4], selecting a set of features for all labels rather than specific features for different labels;
- Selection-LS methods: LLSF [10] and CLML [13];
- Generation-LS methods: ML-LEC [27], LF-LELC [26] and LETTER [6].

For simplicity, we use N1, N2, S1, S2, G1, G2, G3 to represent these algorithms.

For a fair comparison, SVM with RBF kernel is taken as the basic learner to tackle each single-label classification problem. All the hyperparameters of baseline methods are set to the defaults in the corresponding papers, and the hyperparameters of ECL are $m_j = 16$, $\beta = 0.001$.

5.2 Experimental Results

In this subsection, experimental results will be reported. We take 4 well-established multi-label classification metrics to measure the performance, including One Error, Coverage, Ranking Loss and Average Precision [12, 30]. The comparison results between ECL and 7 baseline algorithms are shown in following tables, where the Rank represents the average ranking of an algorithm on 12 datasets.

Table 3. Comparison results on One Error ↓.

DataId	N1	N2	S1	S2	G1	G2	G3	ECL
#1	0.694	0.694	0.786	0.796	0.704	0.765	0.694	0.676
#2	0.339	0.339	0.240	0.240	0.230	0.263	0.224	0.223
#3	0.369	0.379	0.379	0.561	0.379	0.368	0.313	0.337
#4	0.265	0.265	0.265	0.265	0.258	0.277	0.280	0.264
#5	0.457	0.443	0.349	0.375	0.313	0.305	0.291	0.296
#6	0.318	0.331	0.257	0.256	0.199	0.197	0.199	0.196
#7	0.417	0.423	0.277	0.297	0.289	0.280	0.263	0.287
#8	0.298	0.305	0.359	0.349	0.241	0.221	0.211	0.214
#9	0.874	0.871	0.584	0.587	0.569	0.521	0.534	0.534
#10	0.582	0.569	0.569	0.583	0.627	0.537	0.554	0.556
#11	0.246	0.257	0.183	0.183	0.194	0.385	0.183	0.180
#12	0.306	0.302	0.286	0.201	0.187	0.234	0.222	0.208
Average	0.430	0.431	0.378	0.391	0.349	0.363	0.331	0.331
Rank	6.17	6.33	5.08	5.58	4.13	4.00	2.63	2.08

From Tables 3, 4, 5 and 6, it can be seen that:

- LS methods generally achieve better performances comparing to Non-LS methods, implying the effectiveness of modeling on label-specific features;
- In the 4 metrics, ECL obtains the best average ranks, demonstrating its strong abilities in multi-label classification and label correlation exploiting;
- Among the total 48 cases over 12 benchmark datasets and 4 evaluation metrics, ECL ranks first in 15 cases, which outperforms other baselines.

5.3 Ablation Study

In previous work, LIFT [29] employs clustering to generate label-specific features, and LIFTACE [25] incorporates clustering ensemble with cosine label similarity for label correlation exploiting. In this paper, two tips are presented for further improvement, i.e. guiding clustering with entropy and incorporating clustering ensemble with entropy-based label similarity. To verify the effectiveness of these two contributions, an ablation study is conducted by comparing ECL and the following baseline methods:

Table 4. Comparison results on Coverage ↓.

DataId	N1	N2	S1	S2	G1	G2	G3	ECL
#1	0.583	0.583	0.566	0.532	0.532	0.591	0.548	0.522
#2	0.676	0.679	0.626	0.630	0.625	0.658	0.624	0.628
#3	0.506	0.533	0.552	0.698	0.517	0.501	0.509	0.511
#4	0.546	0.521	0.383	0.379	0.386	0.440	0.399	0.384
#5	0.499	0.497	0.400	0.412	0.379	0.378	0.381	0.380
#6	0.271	0.276	0.263	0.262	0.236	0.234	0.236	0.235
#7	0.815	0.823	0.710	0.733	0.696	0.695	0.692	0.698
#8	0.529	0.530	0.698	0.698	0.540	0.578	0.528	0.523
#9	0.275	0.275	0.169	0.156	0.160	0.298	0.155	0.149
#10	0.370	0.362	0.250	0.233	0.245	0.326	0.226	0.228
#11	0.888	0.888	0.840	0.851	0.861	0.914	0.856	0.853
#12	0.763	0.764	0.720	0.648	0.642	0.701	0.650	0.680
Average	0.560	0.561	0.515	0.519	0.485	0.526	0.484	0.483
Rank	6.29	6.96	4.83	4.17	3.50	4.83	2.75	2.67

Table 5. Comparison results on Ranking Loss ↓.

DataId	N1	N2	S1	S2	G1	G2	G3	ECL
#1	0.482	0.439	0.446	0.417	0.401	0.475	0.417	0.393
#2	0.244	0.237	0.212	0.214	0.212	0.249	0.215	0.214
#3	0.364	0.333	0.257	0.418	0.225	0.221	0.215	0.218
#4	0.225	0.208	0.147	0.153	0.160	0.190	0.163	0.156
#5	0.225	0.209	0.182	0.195	0.157	0.155	0.156	0.158
#6	0.120	0.131	0.096	0.095	0.065	0.063	0.066	0.063
#7	0.298	0.307	0.264	0.299	0.250	0.244	0.249	0.250
#8	0.267	0.361	0.345	0.344	0.178	0.191	0.165	0.164
#9	0.181	0.277	0.117	0.106	0.109	0.240	0.104	0.098
#10	0.178	0.176	0.170	0.156	0.167	0.244	0.147	0.148
#11	0.295	0.296	0.209	0.217	0.216	0.309	0.215	0.215
#12	0.241	0.254	0.237	0.177	0.178	0.219	0.189	0.195
Average	0.260	0.269	0.223	0.232	0.193	0.233	0.192	0.189
Rank	6.83	7.00	4.42	4.42	3.17	4.92	2.83	2.42

- Base1: LIFT, determining the number of clusters according to Eq. 2;
- Base2: on the basis of LIFT, determining the number of clusters according to Eq. 10;
- Base3: on the basis of Base2, incorporating clustering ensemble with Cosine similarity; (i.e., on the basis of LIFTACE, determining the number of clusters according to Eq. 10)

The ablation study results are depicted in Fig. 3. Each subbox describes the maximum, upper quartile, median, lower quartile and minimum of the values of

Table 6. Comparison results on Average Precision \uparrow .

DataId	N1	N2	S1	S2	G1	G2	G3	ECL
#1	0.487	0.487	0.455	0.472	0.505	0.450	0.503	0.522
#2	0.759	0.760	0.793	0.790	0.792	0.780	0.792	0.795
#3	0.742	0.727	0.716	0.566	0.736	0.743	0.754	0.748
#4	0.675	0.684	0.752	0.744	0.746	0.721	0.745	0.747
#5	0.782	0.789	0.776	0.756	0.799	0.804	0.805	0.804
#6	0.861	0.853	0.842	0.842	0.882	0.885	0.882	0.885
#7	0.686	0.681	0.680	0.648	0.695	0.704	0.700	0.696
#8	0.736	0.736	0.615	0.616	0.748	0.759	0.772	0.772
#9	0.438	0.467	0.593	0.589	0.599	0.579	0.624	0.629
#10	0.473	0.484	0.599	0.591	0.562	0.585	0.617	0.614
#11	0.389	0.387	0.476	0.461	0.460	0.427	0.465	0.464
#12	0.578	0.581	0.586	0.653	0.645	0.613	0.625	0.632
Average	0.634	0.636	0.657	0.644	0.681	0.671	0.691	0.692
Rank	6.38	6.38	5.00	5.75	3.67	4.50	2.50	1.83

the corresponding metric in 12 datasets. And the circles represet outliers. From the comparison results, we can draw the following conclusion:

- Comparing Base1 and Base2, the ranking of performances of Base2 in four metrics is better than Base1, illustrating the effectiveness of entropy guided clustering.
- ECL significantly outperforms Base3 on all the evaluation metrics, illustrating the effectiveness of entropy-based label similarity guided clustering ensemble.
- Obviously, among the 4 evaluation metrics, the ranking of the performance of ECL and 3 baselines is $ECL \succ Base3 \succ Base2 \succ Base1$. That is, both entropy guided clustering and entropy similarity guided clustering ensemble have made positive contributions, which is in line with our expectation.

5.4 Cluster Entropy Analysis

As analyzed in Sect. 4.1, the first item in the objective function Eq. 10, i.e. weighted sum of cluster entropy (E) is approximately monotonic decreasing with the increasing of cluster number (k).

We perform clustering on all benchmark datasets with k varying from 1 to 100, and obtain the changing values of E . For each dataset, the entropy values for q labels are averaged for a succinct illustration. The relation between E and k is shown in Fig. 4. It can be seen that in every dataset, E keeps a decreasing trend, which is in line with our analysis.

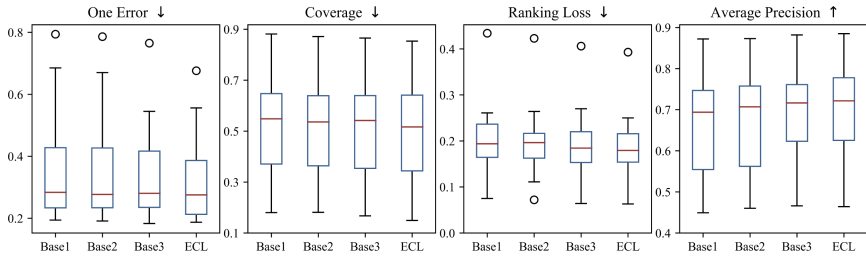


Fig. 3. Ablation study results.

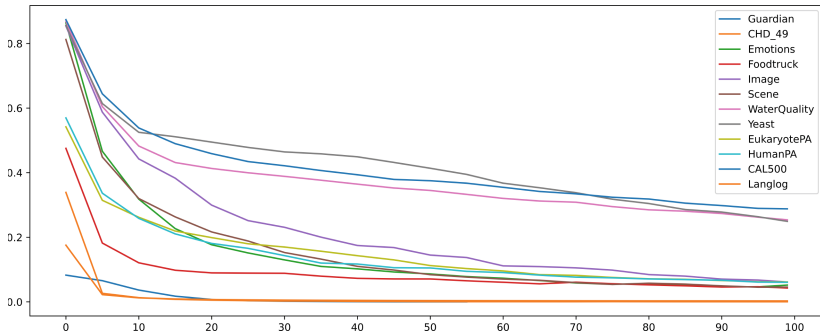


Fig. 4. The relation between E and k .

6 Conclusion

In this paper, a novel label-specific multi-label classification method named ECL is proposed. In ECL, information theory is systematically taken to guide the clustering and clustering ensemble processes. First, a novel definition of cluster entropy is presented to measure the effect of instances distribution on entropy. Second, objective function is established for distinct label to make a quantitative study on the relation between cluster entropy and the number of clusters. Finally, an entropy-based label similarity measurement is designed to consider the distribution information of instances and effectively guides the clustering ensemble. Experimental results on 12 datasets demonstrate that ECL achieves superior performance compared to the 7 baseline methods.

In the future, it is interesting to consider the problem of label-specific features in the deep learning scenarios. Furthermore, class-imbalance problems are common in multi-label classification. It is also meaningful to study class-imbalance problems in label-specific features.

Acknowledgements. This work is supported by the Fundamental Research Funds for the Central Universities (grant xzy022023058) and the National Natural Science Foundation of China (No. 72274152).

References

1. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern recogn.* **37**(9), 1757–1771 (2004)
2. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5177–5186 (2019)
3. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 42–53. Springer (2001)
4. Dai, J., Huang, W., Zhang, C., Liu, J.: Multi-label feature selection by strongly relevant label gain and label mutual aid. *Pattern Recogn.* **145**, 109945 (2024)
5. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. *Adv. Neural. Inf. Process. Syst.* **14**, 681–687 (2001)
6. Guan, Y., Li, W., Zhang, B., Han, B., Ji, M.: Multi-label classification by formulating label-specific features from simultaneous instance level and feature level. *Appl. Intell.* **51**(6), 3375–3390 (2021)
7. Guo, Y., Chung, F., Li, G., Wang, J., Gee, J.C.: Leveraging label-specific discriminant mapping features for multi-label learning. *ACM Trans. Knowl. Discov. Data (TKDD)* **13**(2), 1–23 (2019)
8. Han, H., Huang, M., Zhang, Y., Yang, X., Feng, W.: Multi-label learning with label specific features using correlation information. *IEEE Access* **7**, 11474–11484 (2019)
9. Hang, J.Y., Zhang, M.L.: Collaborative learning of label semantics and deep label-specific features for multi-label classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(12), 9860–9871 (2021)
10. Huang, J., Li, G., Huang, Q., Wu, X.: Learning label specific features for multi-label classification. In: *2015 IEEE International Conference on Data Mining*, pp. 181–190. IEEE (2015)
11. Huang, J., Li, G., Huang, Q., Wu, X.: Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Trans. Knowl. Data Eng.* **28**(12), 3309–3323 (2016)
12. Li, J., Zhu, X., Zhang, W., Wang, J.: A ranking-based problem transformation method for weakly supervised multi-label learning. *Pattern Recogn.* **153**, 110505 (2024)
13. Li, J., Li, P., Hu, X., Yu, K.: Learning common and label-specific features for multi-label classification with correlation information. *Pattern Recogn.* **121**, 108259 (2022)
14. Pan, S., Wu, J., Zhu, X., Long, G., Zhang, C.: Task sensitive feature exploration and learning for multitask graph classification. *IEEE Trans. Cybern.* **47**(3), 744–758 (2016)
15. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 254–269. Springer (2009)
16. Sechidis, K., Tsoumakas, G., Vlahavas, I.: On the stratification of multi-label data. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 145–158. Springer (2011)
17. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)

18. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**(Dec), 583–617 (2002)
19. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: *European Conference on Machine Learning*, pp. 406–417. Springer (2007)
20. Wang, H., et al.: On the value of head labels in multi-label text classification. *ACM Trans. Knowl. Discov. Data* **18**(5), 1–21 (2024)
21. Wang, Y.B., Hang, J.Y., Zhang, M.L.: Stable label-specific features generation for multi-label learning via mixture-based clustering ensemble. *IEEE/CAA J. Automatica Sinica* **9**(7), 1248–1261 (2022)
22. Wei, X., Yu, Z., Zhang, C., Hu, Q.: Ensemble of label specific features for multi-label classification. In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE (2018)
23. Xu, Y.Y., Yang, F., Zhang, Y., Shen, H.B.: An image-based multi-label human protein subcellular localization predictor (i locator) reveals protein mislocalizations in cancer tissues. *Bioinformatics* **29**(16), 2032–2040 (2013)
24. Ye, H., Sunderraman, R., Ji, S.: MatchXML: an efficient text-label matching framework for extreme multi-label text classification. *IEEE Trans. Knowl. Data Eng.* (2024)
25. Zhan, W., Zhang, M.L.: Multi-label learning with label-specific features via clustering ensemble. In: *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 129–136. IEEE (2017)
26. Zhang, C., Li, Z.: Multi-label learning with label-specific features via weighting and label entropy guided clustering ensemble. *Neurocomputing* **419**, 59–69 (2021)
27. Zhang, J.J., Fang, M., Li, X.: Multi-label learning based on label entropy guided clustering. In: *2014 IEEE International Conference on Computer and Information Technology*, pp. 756–760. IEEE (2014)
28. Zhang, J.J., Fang, M., Li, X.: Multi-label learning with discriminative features for each label. *Neurocomputing* **154**, 305–316 (2015)
29. Zhang, M.L., Wu, L.: Lift: multi-label learning with label-specific features. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(1), 107–120 (2014)
30. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recogn.* **40**(7), 2038–2048 (2007)
31. Zhang, Y., Huo, W., Tang, J.: Multi-label feature selection via latent representation learning and dynamic graph constraints. *Pattern Recogn.* **151**, 110411 (2024)
32. Zhu, K., Fu, M., Wu, J.: Multi-label self-supervised learning with scene images. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6694–6703 (2023)
33. Zhu, X., Lu, W.: Multi-label classification with dual tail-node augmentation for drug repositioning. *IEEE/ACM Trans. Comput. Biol. Bioinform.* (2023)



FAT-LSTM: A Multimodal Data Fusion Model with Gating and Attention-Based LSTM for Time-Series Classification

Pouya Hosseinzadeh^(✉), Omar Bahri, Soukaina Filali Boubrahimi,
and Shah Muhammad Hamdi

Department of Computer Science, Utah State University, Logan, UT, USA
{pouya.hosseinzadeh,omar.bahri,soukaina.boubrahimi,s.hamdi}@usu.edu

Abstract. The remarkable success of machine learning models has sparked considerable interest in multimodal data fusion techniques. Addressing the challenge of integrating diverse data modalities while enhancing classification performance remains a key focus. In this paper, we introduce FAT-LSTM, a novel framework designed to fuse multimodal data for classification tasks. Leveraging advanced Gating and Attention-based Long Short-Term Memory (LSTM) mechanisms, FAT-LSTM represents a significant improvement in multimodal learning field. Through extensive comparative analysis against established baseline models, our study shows the superior performance achieved by FAT-LSTM. Additionally, thorough ablation analysis provides insights into the inner workings of the model, shedding light on its effectiveness. Empirical validation across multiple space weather datasets further confirms FAT-LSTM's efficacy across various scenarios. This research highlights the crucial role of multimodal data fusion in effectively addressing real-world challenges.

Keywords: Multimodal Classification · Time Series Analysis · Data Fusion · Deep Learning · Space Weather

1 Introduction

In today's data-driven landscape, the exponential growth of data across diverse domains has given rise to innovative research directions in data analysis and classification. A particularly intriguing advancement is the emergence of multimodal data fusion for time series classification. Unlike conventional approaches that focus on a single data source, multimodal data fusion integrates insights from multiple sources or modalities, offering the potential to enhance classification accuracy and provide a comprehensive understanding of underlying patterns within time series data [1, 4, 5].

Time series data, characterized by sequential observations over time, plays a pivotal role in various domains such as finance, healthcare, environmental monitoring, and industrial processes. Accurate analysis and classification of time series data are critical for tasks ranging from anomaly detection to predictive

maintenance. However, traditional time series classification methods often rely solely on the temporal information present in a single modality, which might be limiting when dealing with complex, heterogeneous, and noisy data.

The concept of multimodal data fusion revolves around the notion that distinct modalities—ranging from visual and textual to temporal—can offer complementary and diverse insights into the same phenomenon [9, 24]. Integrating information from multiple modalities aims to achieve a more comprehensive and accurate data representation. This approach not only helps capture intricate patterns that could elude single-modal methods but also enhances robustness against noise and uncertainties inherent in individual modalities.

In the realm of space weather forecasting, the fusion of multimodal data is offering new horizons for predicting Solar Energetic Particle (SEP) events. These rare but impactful solar events, triggered by solar flares and coronal mass ejections, pose significant threats to space-based technologies [6–8]. By integrating data from diverse sources—ranging from solar observations to magnetic field measurements—multimodal fusion provides a comprehensive view of the intricate solar dynamics that precede SEP events. This approach capitalizes on both temporal and spatial patterns, resulting in improved event prediction accuracy. The ongoing challenges in data integration, formulating effective fusion strategies, and achieving interpretability reflect the dynamic nature of this field. By leveraging multimodal data fusion, we aim to enhance space weather prediction and improve our ability to manage the unpredictable effects of the space environment.

As we explore the data fusion field, various techniques merge information from multiple sources to extract comprehensive insights. These techniques include early fusion, late fusion, and decision-level fusion, each with distinct strengths and applications. In this context, our paper introduces an innovative approach—a novel attention-based model—for multimodal data classification, known as FAT-LSTM. This novel model uses attention mechanisms to effectively combine modalities, enhancing classification accuracy while unveiling intricate relationships within the data. By proposing this method, we contribute to the evolution of data fusion methodologies, striving for more effective and insightful multimodal data analysis.

To show the superiority of our proposed method, we conduct experiments comparing our model with baseline methods. The evaluation is performed on three benchmark datasets of SEP rare events from the solar weather prediction domain. The contribution of this work is twofold. Firstly, we present a novel multimodal data fusion technique that advances the field of time series classification. Secondly, our work significantly contributes to the improvement of SEP event prediction. By utilizing the power of multimodal data fusion, we enhance the accuracy and reliability of SEP event forecasts.

While this study primarily focuses on space weather prediction to rigorously validate the FAT-LSTM model under controlled conditions, we recognize the importance of showing the model’s resilience and usefulness across various domains. Future research will explore the application of FAT-LSTM in diverse

fields such as healthcare, finance, and environmental monitoring, showcasing its broad relevance and impact.

The rest of this paper is organized as follows: Sect. 2 outlines background and related works. Section 3 gives a thorough outline of the methodology utilized in this research. In Sect. 4, we detail the experimental arrangement, and the resulting discoveries. Lastly, Sect. 5 encapsulates our findings and deliberates on potential paths for future research endeavors.

2 Related Works

Early multimodal classification studies focused on combining features from different sources using traditional techniques to enhance accuracy in tasks like sentiment analysis, image captioning, and speech recognition. These methods faced challenges in capturing complex relationships between modalities due to heterogeneous data characteristics and high computational complexity. Common approaches included early fusion, which concatenates raw feature vectors from various modalities, and late fusion, which aggregates classification results from individual classifiers. [2, 3]. These methods, although effective to some extent, often faced difficulties in maintaining the balance between computational efficiency and the richness of the integrated features and effectively capturing complex relationships between modalities. We benchmarked our results by comparing them with those presented in a relevant review paper that comprehensively summarizes data fusion techniques by Sleeman et al. [10].

The advent of deep learning introduced novel paradigms for multimodal data fusion, offering capabilities to model intricate interactions between different modalities. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were adapted to process modalities such as images and text, respectively [11, 12, 15]. Researchers also proposed architectures specifically designed for fusing information from different sources. For instance, attention mechanisms emerged as a powerful tool for selectively attending to relevant features across modalities. These mechanisms allowed for dynamic fusion, enabling models to focus on pertinent information for improved classification performance [13, 14].

To address challenges associated with heterogeneous modalities, cross-modal embeddings gained traction. These embeddings aimed to project data from different sources into a shared latent space, facilitating more effective fusion. Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) were also employed to generate missing modalities, aiding in scenarios where certain data sources were incomplete or unavailable. These generative models not only enhanced the completeness of data but also improved the robustness of classification systems [16].

In addition to the aforementioned techniques, recent developments in multimodal data fusion have expanded the horizons of this field. One notable area of research is the integration of multimodal data for healthcare applications. In medical diagnosis and treatment planning, combining data from sources like

medical images, patient records, and textual reports has become crucial. Multimodal fusion approaches in healthcare aim to improve disease prediction, treatment selection, and patient care by harnessing the complementary information available in various modalities [17–19].

In addition to these techniques, another significant approach in multimodal data fusion is Graph-based Fusion. Graph-based methods model the relationships between different modalities as a graph, where nodes represent modalities, and edges denote the relationships between them. By leveraging graph-based techniques, researchers can capture intricate dependencies and correlations between modalities, leading to more accurate and robust multimodal classification [21].

Another compelling avenue in multimodal data fusion is its application in autonomous systems and robotics. Robots equipped with sensors that capture data from visual, auditory, and tactile modalities can benefit greatly from multimodal fusion. Combining information from these sensors enables robots to make more informed decisions in navigation, object recognition, and human-robot interaction. This field presents unique challenges in aligning data streams from disparate sensors, and researchers have been exploring innovative techniques to tackle these challenges effectively [21].

3 Methodology

In this section, we explain the data preprocessing, and then we present the proposed FAT-LSTM method for multimodal data fusion in classification tasks. We describe the architecture and components of FAT-LSTM in detail and outline the steps involved in its operation. Furthermore, we discuss the baseline methods that we selected for comparison and detail how we conducted the comparative evaluation. Detailed information about data preprocessing and the source code for our experiments can be accessed on our project website: <https://sites.google.com/view/mtsf-cpf/home>.

3.1 Data Preprocessing

In this research, we employed fusion-based multimodal classifiers, which were trained using two distinct types of data: time series data and image data. The purpose of the image data was to represent rare solar energetic particle events. These datasets were divided into three subsets, namely SolEnergiNet-100 for 100 MeV data, SolEnergiNet-60 for 60 MeV data, and SolEnergiNet-30 for 30 MeV data. For detailed dataset information, refer to Table 1, which provides the relevant metadata. In our research, modality M1 corresponds to the time series 5-hour proton flux data from the Sun, gathered from the Geostationary Operational Environmental Satellite (GOES) satellite, and is publicly accessible at: <https://www.ncei.noaa.gov/data/goes-space-environment-monitor/access/avg/>. The second modality, M2, consists of single-shot images of the Sun captured by SOHO spacecraft, available at <https://helioviewer.org/>. To

convert the images from modality M2 into a vector representation, we employed an autoencoder, a neural network architecture composed of two main components: an encoder and a decoder.

Table 1. Datasets Metadata

Dataset Name	Labels	Time Series Length	Image Vector Length	Dataset Size
SolEnergiNet-100	2	60 (5h)	60	96
SolEnergiNet-60	2	60 (5h)	60	40
SolEnergiNet-30	2	60 (5h)	60	44

During the training phase, the encoder takes the input images and processes them through a series of hidden layers, reducing the images to a lower-dimensional representation often referred to as the “latent space” or “encoded space”. This latent space captures essential features and patterns from the original images. Subsequently, the decoder takes these lower-dimensional representations and enlarges them, reconstructing the images to match the original input size. The primary objective of the autoencoder is to minimize the difference between the original images and their reconstructed counterparts, effectively learning to compress and then reconstruct the data. Once the autoencoder is trained on a substantial set of images, we can utilize the output of the encoder component (the hidden layer) as the vector representation (X2) for each image [22, 23]. This encoder output provides a concise and meaningful representation of the image, capturing its significant features in a reduced-dimensional space. It is important to note that all three datasets in this study include binary labels indicating the presence or absence of a solar event. The first time series modality, M1, has a fixed length consisting of 60 time steps. In contrast, the second modality (M2), which is a single-shot image, was transformed into a low-dimensional vector comprising 60 dimensions. It is worth emphasizing that solar events are infrequent occurrences. Consequently, the dataset sizes reflect this rarity, with SolEnergiNet-100 containing 96 instances, SolEnergiNet-60 comprising 40 instances, and SolEnergiNet-30 consisting of 44 instances.

3.2 FAT-LSTM

The FAT-LSTM architecture consists of several key components, each contributing to the process of multimodal data fusion. Refer to Algorithm 1 for a detailed step-by-step representation of the FAT-LSTM process. In addition, Fig. 1 displays the FAT-LSTM network architecture for multimodal time series classification. The process involves taking a number of multimodal data as input and then feeding them to the network layers. Furthermore, the following part delves into the mathematical framework underpinning the FAT-LSTM methodology, clarifying the equations that define its distinct components. These equations present a structured account of how FAT-LSTM adeptly integrates multimodal

data. It is important to note that we define the input data modalities as X instead of M , representing the outcome of preprocessing on input data, such as the transformation of images into new feature vectors.

Algorithm 1. FAT-LSTM

Require: $X = \{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\}$ \triangleright Input time-series data and labels
Require: \mathbb{R}^T \triangleright Feature space

```

1: function FAT_LSTM( $X$ )
2:   Initialize empty arrays:  $LSTM\_gated, Att$ 
3:   for  $i = 1$  to  $n$  do
4:      $LSTM_i \leftarrow LSTM(X_i)$   $\triangleright$  Apply LSTM to the  $i$ -th input
5:      $Gating_i \leftarrow \sigma(\text{Dense}(LSTM_i))$   $\triangleright$  Calculate gating factor for  $i$ -th input
6:      $LSTM_{gated,i} \leftarrow LSTM_i \odot Gating_i$   $\triangleright$  Apply gating to LSTM output
7:      $LSTM\_gated.append(LSTM_{gated,i})$ 
8:   end for
9:   for  $i = 1$  to  $n$  do
10:     $Att_i \leftarrow \text{Self-Attention}(LSTM_{gated,i})$   $\triangleright$  Apply self-attention to gated LSTM
        output
11:     $Att.append(Att_i)$ 
12:   end for
13:    $Concatenated \leftarrow \text{Concatenate}(Att_1, Att_2, \dots, Att_n)$   $\triangleright$  Concatenate attended
        representations
14:    $Fusion \leftarrow \text{ReLU}(\text{Dense}(Concatenated))$   $\triangleright$  Apply ReLU to fused
        representation
15:   Output  $\leftarrow \text{Softmax}(Fusion)$   $\triangleright$  Apply softmax for classification
16:   return Output
17: end function

```

1. **LSTM Blocks:** For each modality i , the input data X is passed through an LSTM [25] block, producing an intermediate output $LSTM_i$. The LSTM cell's operations are as follows:

$$f_t^{(i)} = \sigma(W_f^{(i)} \cdot [h_{t-1}^{(i)}, x_t^{(i)}] + b_f^{(i)}) \quad (1)$$

$$g_t^{(i)} = \sigma(W_i^{(i)} \cdot [h_{t-1}^{(i)}, x_t^{(i)}] + b_i^{(i)}) \quad (2)$$

$$\tilde{C}_t^{(i)} = \tanh(W_C^{(i)} \cdot [h_{t-1}^{(i)}, x_t^{(i)}] + b_C^{(i)}) \quad (3)$$

$$C_t^{(i)} = f_t^{(i)} \odot C_{t-1}^{(i)} + g_t^{(i)} \odot \tilde{C}_t^{(i)} \quad (4)$$

$$o_t^{(i)} = \sigma(W_o^{(i)} \cdot [h_{t-1}^{(i)}, x_t^{(i)}] + b_o^{(i)}) \quad (5)$$

$$h_t^{(i)} = o_t^{(i)} \odot \tanh(C_t^{(i)}) \quad (6)$$

The weights $W_f^{(i)}, W_i^{(i)}, W_C^{(i)}, W_o^{(i)}$ and biases $b_f^{(i)}, b_i^{(i)}, b_C^{(i)}, b_o^{(i)}$ are specific to each modality i and are not shared among different modalities.

2. **Gating Mechanism:** The output of each modality-specific LSTM block is fed into a dense layer with a sigmoid activation function, denoted as $Gating_i$. This gating mechanism controls the relevance of each modality's information:

$$Gating_i = \sigma(W_{gate}^{(i)} \cdot h^{(i)} + b_{gate}^{(i)}) \quad (7)$$

3. **Element-wise Multiplication:** The output of the LSTM block is multiplied element-wise with its corresponding gating factor:

$$LSTM_{gated}^{(i)} = h^{(i)} \odot Gating_i \quad (8)$$

4. **Self-Attention Mechanism:** The gated LSTM outputs for all modalities are passed through a self-attention mechanism, resulting in a set of attended representations Att_i . The attention mechanism's operations are as follows:

$$q^{(i)} = W_q^{(i)} \cdot LSTM_{gated}^{(i)} \quad (9)$$

$$k^{(i)} = W_k^{(i)} \cdot LSTM_{gated}^{(i)} \quad (10)$$

$$v^{(i)} = W_v^{(i)} \cdot LSTM_{gated}^{(i)} \quad (11)$$

$$\text{Attention}^{(i)} = \text{softmax} \left(\frac{q^{(i)} \cdot k^{(i)\top}}{\sqrt{d_k}} \right) \cdot v^{(i)} \quad (12)$$

$$\text{Att}_h^{(i)} = \text{Attention}^{(i)} + LSTM_{gated}^{(i)} \quad (13)$$

The weights $W_q^{(i)}, W_k^{(i)}, W_v^{(i)}$ are specific to each modality i , ensuring that the attention mechanism can focus on modality-specific features.

5. **Concatenation:** The attended representations from all modalities are concatenated together:

$$\text{Concatenated} = \text{concat}(Att_1, Att_2, \dots, Att_n) \quad (14)$$

6. **Fusion Layer:** The concatenated representation is passed through a dense layer with ReLU activation, denoted as $Fusion$:

$$Fusion = \text{ReLU}(W_{fusion} \cdot \text{Concatenated} + b_{fusion}) \quad (15)$$

7. **Classification:** The fused representation is further processed by a softmax layer for classification:

$$\text{Output} = \text{Softmax}(W_{softmax} \cdot Fusion + b_{softmax}) \quad (16)$$

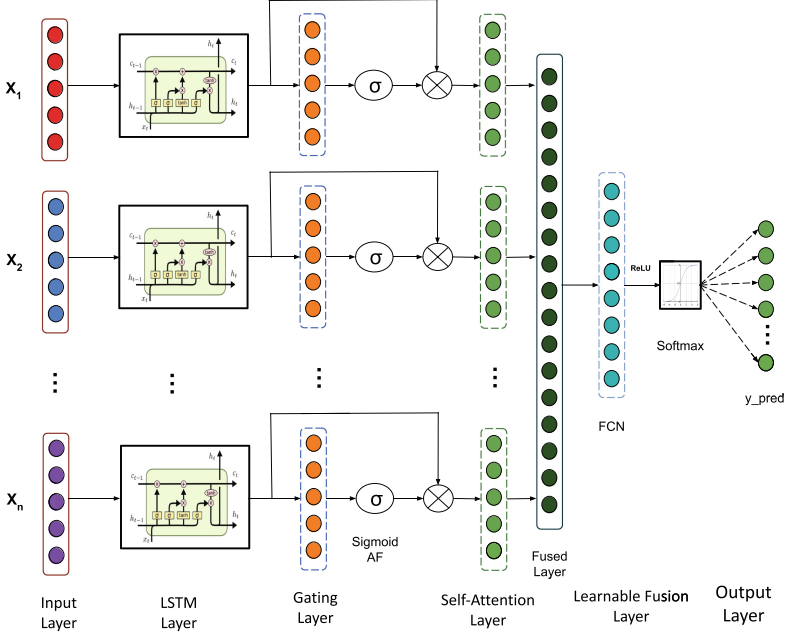


Fig. 1. FAT-LSTM network architecture for multimodal time series classification.

3.3 Baseline Methods

We compared our proposed FAT-LSTM multimodal model with four other baselines within the fusion category as outlined in [10], as well as with two state-of-the-art approaches using VAE and CNN models.

- **Feature Concatenation (FC):** concatenates the features from different modalities into a single vector representation [26,27]. The concatenated feature vector is then used as input to a classification model.
- **Deep Concatenation (DC):** also known as Deep Feature Concatenation, combines features from different modalities at a deep representation level. The latter involves feeding the data from different modalities into separate deep neural networks and concatenating the high-level representations learned by the networks before making a final prediction [28]. DC allows the model to capture complex interactions and dependencies between modalities.
- **Deep Merge (DM):** combines features from different modalities at a lower level in the network, typically through element-wise operations like summation or multiplication [29]. This approach aims to capture both shared and unique information from each modality.
- **Score Concatenation (SC):** involves training separate models for each modality. The prediction scores of the individual models are concatenated and used as input to a final classification model [17]. This approach leverages the strengths of each modality-specific model while combining their predictions

to make a final decision. An example of the Score Concatenation technique is ensemble learning.

- **Variational Autoencoder (VAE)**: uses an encoder-decoder architecture where the encoder compresses the input data into a probabilistic latent space, and the decoder reconstructs the input from this latent space. The encoded representations are used as features for classification.
- **Convolutional Neural Network (CNN)**: applies convolutional layers to extract spatial features from the input data. The extracted features are then used for classification.

Figure 2 illustrates the four fusion baselines (FC, DC, DM, SC) used to compare our method with. The additional models (VAE, CNN) were included to further evaluate the robustness of our approach with other deep learning models.

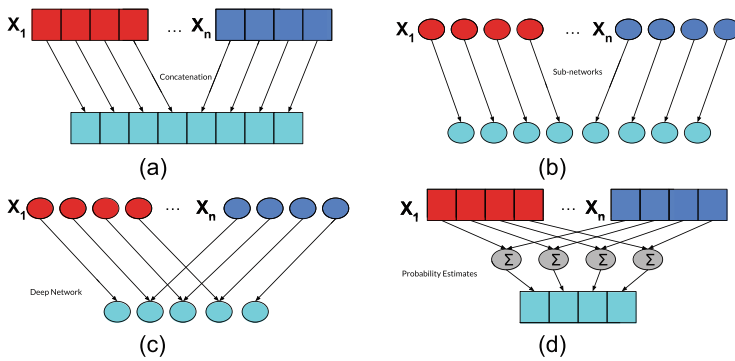


Fig. 2. Architecture of (a) Feature Concatenation, (b) Deep Concatenation, (c) Deep Merge, and (d) Score concatenation data fusion methods within the fusion category as outlined in [10].

4 Experiments

Within this section, we first lay the ground for our experimental setup and present the experimental results obtained after training FAT-LSTM and the other multimodal fusion techniques discussed in Sect. 3.3.

4.1 Experimental Setup

The following parameters represent the optimal hyper-parameters for SolEnergiNet-100 dataset: 5-fold stratified cross-validation to ensure robust evaluation, standard scaling for feature normalization, utilization of two LSTM layers with 128 units for each input modality, integration of a gating mechanism

with sigmoid activation within the LSTM layers, incorporation of self-attention mechanisms to capture intra-modality relationships, computation of weighted sums through attention-weighted outputs, inclusion of a fusion layer (Dense, 256 units, ReLU activation) for cross-modality integration, utilization of an output layer (Dense, 2 units, softmax activation) for binary classification, optimization through the Adam optimizer with default learning rate, and training conducted over 15 epochs with a batch size of 16. The other two datasets were trained and evaluated using nearly identical hyper-parameters.

4.2 Performance Measures

To evaluate the performance of our model, we conducted several experiments. We used standard evaluation metrics, including Accuracy, F1-score, Precision, Recall, True Skill Statistics (TSS), and updated Heidke skill score (HSS2) [30]. TSS and HSS2 metrics are used to assess the skill of categorical predictions or forecasts, particularly in meteorology and climatology. Equations 17 and 18 define TSS and HSS2, respectively.

$$TSS = \frac{TP}{TP + FN} - \frac{FP}{FP + TN} \quad (17)$$

$$HSS2 = \frac{2 * ((TP * TN) - (FN * FP))}{(TP + FN) * (FN + TN) + (FP + TN) * (TP + FP)} \quad (18)$$

where, TP, FP, TN, and FN represent True Positive, False Positive, True Negative, and False Negative, respectively.

We used the average of a 5-fold cross-validation technique to report our results. This approach allowed us to assess the performance of all the baseline data fusion techniques using different subsets of the data.

4.3 Experimental Results

In this section, we present the comprehensive performance evaluation of our novel model, FAT-LSTM, in comparison to other models, utilizing a 5-fold cross-validation strategy. Our analysis encompasses standard evaluation metrics to provide a holistic perspective on the model's performance across different datasets, as presented in Table 2.

FAT-LSTM consistently demonstrates remarkable performance in all measured aspects, emphasizing its effectiveness as a classification model. Regardless of the dataset, it consistently outperforms other models across all evaluation metrics.

On the SolEnergiNet-100 dataset, FAT-LSTM stands out with an impressive accuracy of 87%, underscoring its capability to make accurate predictions. This level of precision is mirrored in its F1-score, where FAT-LSTM achieves a matching 87%, showcasing its balance between precision and recall.

Moreover, FAT-LSTM excels in terms of precision, especially highlighted on the SolEnergiNet-60 dataset with an exceptional precision score of 87%. This precision signifies FAT-LSTM's proficiency in making precise positive classifications. In evaluating recall, FAT-LSTM consistently maintains high recall rates, indicating its ability to effectively capture positive instances in the dataset. On SolEnergiNet-60, FAT-LSTM shares the highest recall rate of 86%. Even on the challenging SolEnergiNet-30 dataset, FAT-LSTM stands out with a recall of 70%.

Furthermore, FAT-LSTM shows its strong ability to differentiate between classes with a high TSS on all datasets. For SolEnergiNet-100, it achieves a TSS of 74%, signifying its strong ability to discriminate between classes. Similarly, on SolEnergiNet-60, FAT-LSTM maintains a TSS of 76%, reflecting its capacity to distinguish between data points effectively. On SolEnergiNet-30, FAT-LSTM exhibits a competitive TSS of 41%. However, FC has the same TSS score. HSS2 also highlights FAT-LSTM's overall classification skill. Across all datasets, it maintains a lead in HSS2. On SolEnergiNet-100, FAT-LSTM achieves an HSS2 of 74%, showcasing its overall skill in classifying data. In the case of SolEnergiNet-60, FAT-LSTM maintains a high HSS2 of 75%, further confirming its strong classification performance. While excelling in various metrics, FAT-LSTM does not outperform FC in terms of HSS2 on SolEnergiNet-30 dataset.

In summary, FAT-LSTM consistently outperforms other models across various datasets and evaluation metrics, demonstrating its effectiveness as a classification model. These results underscore FAT-LSTM's potential as a valuable tool in a wide range of applications where precise and reliable classification is paramount. Additionally, the poor performance of the VAE and CNN models indicates that these approaches may not be well-suited for space weather data, especially when the dataset is limited in size. Further investigations and fine-tuning may be explored to maximize FAT-LSTM's impact in specific domains.

4.4 Ablation Study

We conducted an ablation analysis to investigate the individual contributions of different components within our model, FAT-LSTM, towards its outstanding performance. This analysis involved the creation of two distinct models by selectively removing specific components from FAT-LSTM, resulting in AT-LSTM (Attention-based LSTM without gating component and dense fusion layer) and LSTM models. Therefore, our multimodal LSTM model does not include gating or self-attention layers, while the AT-LSTM model lacks a gating layer. In contrast, the FAT-LSTM model incorporates both gating and self-attention layers, as shown in Fig. 1. The ablation analysis aimed to dissect the contributions of each component in FAT-LSTM and understand how these elements collectively influence its classification process.

As Fig. 3 displays, these percentages signify the extent of performance variations between FAT-LSTM and the alternative models, AT-LSTM and LSTM. For instance, the notable 10.1% difference in F1-score underscores FAT-LSTM's substantial improvement in achieving a balanced trade-off between precision and

Table 2. Average performance results of 5-fold cross-validation. Values for winning models are shown in bold, while runner-up models are indicated by underscores.

Dataset	Model	Metric					
		Accuracy	F1-score	Precision	Recall	TSS	HSS2
SolEnergiNet-100	FC	82	<u>82</u>	<u>83</u>	<u>82</u>	<u>65</u>	<u>65</u>
	DC	75	75	75	75	50	50
	DM	76	76	76	76	52	52
	SC	77	77	77	77	54	54
	VAE	67	67	67	67	35	35
	CNN	70	69	73	70	41	41
	FAT-LSTM	87	87	88	87	74	74
SolEnergiNet-60	FC	85	85	86	<u>86</u>	<u>71</u>	<u>71</u>
	DC	80	79	81	80	59	59
	DM	<u>86</u>	<u>86</u>	<u>87</u>	<u>86</u>	<u>71</u>	70
	SC	80	78	84	80	59	59
	VAE	80	79	82	80	61	60
	CNN	65	63	67	64	29	29
	FAT-LSTM	87	87	90	88	76	75
SolEnergiNet-30	FC	<u>69</u>	<u>69</u>	66	70	41	41
	DC	59	59	59	59	18	18
	DM	66	66	<u>72</u>	<u>67</u>	<u>34</u>	34
	SC	62	62	65	62	21	22
	VAE	59	49	47	58	17	16
	CNN	57	53	65	58	16	17
	FAT-LSTM	70	70	74	70	41	<u>40</u>

recall compared to the other models. These percentages provide a quantitative perspective on the performance gaps. Precision percentages indicate that FAT-LSTM consistently maintained significantly higher precision in its predictions, highlighting the role of the gating and fusion layer’s in accurate positive classifications. Recall percentages emphasize FAT-LSTM’s excellence in capturing positive instances in the dataset, with the presence of gating, fusion, and self-attention layers contributing to this achievement. Moreover, the percentages for HSS2 underscore that FAT-LSTM exhibited superior overall classification skill, largely driven by the combined effects of gating, fusion, and self-attention layers. These percentage differences not only quantify the performance disparities but also reaffirm the critical role of the gating, fusion, and self-attention mechanisms in enhancing the model’s classification capabilities.

In our pursuit of a comprehensive understanding of model performance and efficiency, we conducted another ablation analysis with a focus on sequence analysis and runtimes. This analysis aimed to shed light on the impact of varying

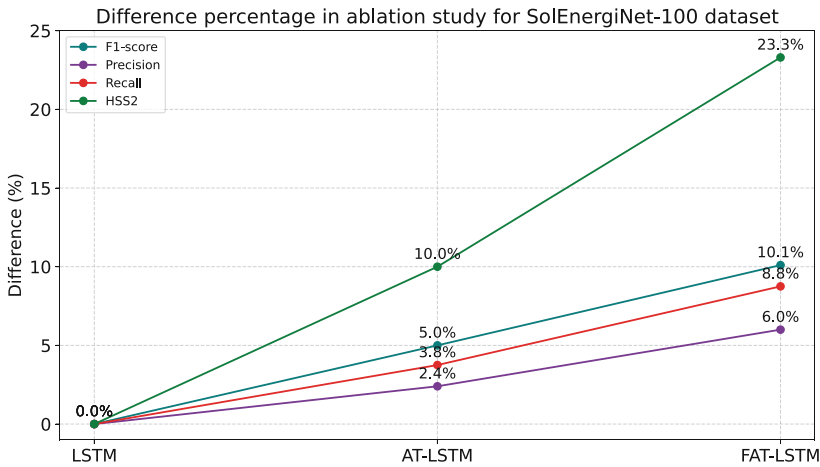


Fig. 3. Connected scatter plot with difference percentage obtained from ablation study performed on FAT-LSTM using SolEnergiNet-100 dataset.

observation windows (15h, 12h, 9h, 6h, and 3h) of the time series proton flux data modality on the performance of three distinct models: FAT-LSTM, AT-LSTM, and LSTM. By thoroughly exploring these different observation window sizes, we gained valuable insights into how the temporal context influences the models' classification performance and computational efficiency. The observation window captures the critical period leading up to the onset of a solar flare event, which is important as a solar flare event may lead to an SEP event.

The performance evaluation, as shown in the bar plots of Fig. 4, centered around the HSS2. Each observation window's HSS2 values for the three models were carefully scrutinized. These HSS2 scores provide an insightful perspective on the models' classification skills under different time constraints. Across varying observation windows, FAT-LSTM consistently shows competitive HSS2 scores. This consistency in performance underscores the model's robustness in handling diverse temporal contexts. Notably, FAT-LSTM outperformed both AT-LSTM and LSTM in several instances, reaffirming its superiority in maintaining high classification skills. Complementing the performance assessment, our analysis delved into runtime considerations. The line plots in Fig. 4 portray the runtime data associated with each model across different observation windows. The runtimes for FAT-LSTM, AT-LSTM, and LSTM were tracked and plotted, providing valuable insights into the computational efficiency of these models. Intriguingly, the runtime analysis revealed that FAT-LSTM maintained a favorable balance between classification performance and computational efficiency. Despite its impressive performance, FAT-LSTM exhibited runtimes comparable to or even more efficient than the alternative models. This observation is particularly notable when considering the shorter 3-hour observation window, where computational efficiency becomes essential.

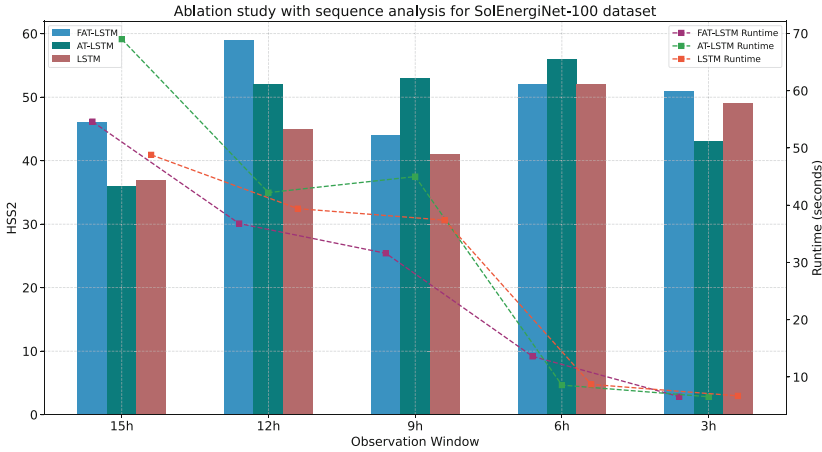


Fig. 4. Evaluation of different variants of LSTM including FAT-LSTM for SolEnergiNet-100 dataset.

5 Conclusion

In this study, we introduced FAT-LSTM (A Multimodal Data Fusion Model with Gating and Attention-Based LSTM for Classification), a novel approach designed to enhance classification performance by effectively fusing information from diverse modalities. By integrating gating and attention mechanisms into the LSTM architecture, FAT-LSTM exhibited significant improvements over established baseline models. We also conducted ablation analysis to gain insights into the specific contributions of gating and attention mechanisms, further elucidating the model's behavior. The empirical validation of FAT-LSTM on space weather datasets showcased its efficacy and versatility, yet its generalizability to other domains remains a topic for future research. Additionally, interpreting how FAT-LSTM combines information from various sources and addressing potential scalability challenges are areas of ongoing investigation. In conclusion, FAT-LSTM represents a promising advancement in multimodal data fusion for classification tasks, but its broader applicability, interpretability, and scalability deserve further exploration. This work underscores the potential of such models to address real-world challenges effectively and encourages continued research in this direction.

Acknowledgment. This project has been supported in part by funding from the Division of Atmospheric and Geospace Sciences within the Directorate for Geosciences, under NSF awards #2301397, #2204363, and #2240022, and by funding from the Office of Advanced Cyberinfrastructure within the Directorate for Computer and Information Science and Engineering, under NSF award #2305781.

References

1. Archila, J., Manzanera, A., Martinez, F.: A multimodal Parkinson quantification by fusing eye and gait motion patterns, using covariance descriptors, from non-invasive computer vision. In: *Computer Methods and Programs in Biomedicine*, 2015, pp. 106607 (2015)
2. Gunes, H., Piccardi, M.: Affect recognition from face and body: early fusion vs. late fusion. In: *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3437–3443. IEEE (2005)
3. Gadzicki, K., Khamsehashari, R., Zetzsche, C.: Early vs late fusion in multimodal convolutional neural networks. In: *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–6. IEEE (2020)
4. Chen, Y., Kempton, D.J., Ahmadzadeh, A., Angryk, R.A.: Towards synthetic multivariate time series generation for flare forecasting. In: Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds.) *ICAISC 2021. LNCS (LNAI)*, vol. 12854, pp. 296–307. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87986-0_26
5. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. *Inf. Sci.* **239**, 142–153 (2013)
6. Whitman, K., et al.: Review of solar energetic particle models. In: *Advances in Space Research*, Elsevier (2022)
7. Hosseinzadeh, P., Boubrahimi, S.F., Hamdi, S.M.: Improving solar energetic particle event prediction through multivariate time series data augmentation. *Astrophys. J. Suppl. Ser.* **270**(2), 31 (2024). IOP Publishing
8. Hosseinzadeh, P., Filali Boubrahimi, S., Hamdi, S.M.: Toward enhanced prediction of high-impact solar energetic particle events using multimodal time series data fusion models. *Space Weather* **22**(6), e2024SW003982 (2024)
9. Filali Boubrahimi, S., Neema, A., Nassar, A., Hosseinzadeh, P., Hamdi, S.M.: Spatiotemporal data augmentation of MODIS-landsat water bodies using adversarial networks. *Water Resour. Res.* **60**(3), e2023WR036342 (2024)
10. Sleeman, W.C., IV., Kapoor, R., Ghosh, P.: Multimodal classification: current landscape, taxonomy and future directions. *ACM Comput. Surv.* **55**(7), 1–31 (2022)
11. Dolz, J., Gopinath, K., Yuan, J., Lombaert, H., Desrosiers, C., Ayed, I.B.: HyperDense-Net: a hyper-densely connected CNN for multi-modal image segmentation. *IEEE Trans. Med. Imaging* **38**(5), 1116–1126 (2018)
12. Shenoy, A., Sardana, A.: Multilogue-net: A context-aware RNN for multimodal emotion detection and sentiment analysis in conversation. *arXiv preprint arXiv:2002.08267* (2020)
13. Zhang, X., Liang, X., Zhiyuli, A., Zhang, S., Xu, R., Wu, B.: At-LSTM: An attention-based LSTM model for financial time series prediction. In: *IOP Conference Series: Materials Science and Engineering*, vol. 569, no. 5, pp. 052037. IOP Publishing (2019)
14. Wang, Y., et al.: Adversarial multimodal fusion with attention mechanism for skin lesion classification using clinical and dermoscopic images. *Med. Image Anal.* **81**, 102535 (2022). Elsevier
15. EskandariNasab, M., Raeisi, Z., Lashaki, R.A., Najafi, H.: A GRU-CNN model for auditory attention detection using microstate and recurrence quantification analysis. *Sci. Rep.* **14**(1), 8861 (2024)
16. Suzuki, M., Nakayama, K., Matsuo, Y.: Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891* (2016)

17. Guggenmos, M., et al.: A multimodal neuroimaging classifier for alcohol dependence. *Sci. Rep.* **10**(1), 298 (2020)
18. Higuera, C., Gardiner, K.J., Cios, K.J.: Self-organizing feature maps identify proteins critical to learning in a mouse model of Down syndrome. *PLoS ONE* **10**(6), e0129126 (2015)
19. Singh, R., Khare, A.: Fusion of multimodal medical images using Daubechies complex wavelet transform-a multiresolution approach. *Inf. Fusion* **19**, 49–60 (2014)
20. Weerakoon, K., Sathyamoorthy, A.J., Liang, J., Guan, T., Patel, U., Manocha, D.: GrASPE: Graph based multimodal fusion for robot navigation in unstructured outdoor environments. *arXiv preprint [arXiv:2209.05722](https://arxiv.org/abs/2209.05722)* (2022)
21. Kubelka, V., Reinstein, M., Svoboda, T.: Improving multimodal data fusion for mobile robots by trajectory smoothing. *Robot. Auton. Syst.* **84**, 88–96 (2016). Elsevier
22. Öztürk, Ş.: Stacked auto-encoder based tagging with deep features for content-based medical image retrieval. *Expert Syst. Appl.* **161**, 113693 (2020). Elsevier
23. Thomas, S.A., Race, A.M., Steven, R.T., Gilmore, I.S., Bunch, J.: Dimensionality reduction of mass spectrometry imaging data using autoencoders. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2016, pp. 1–7 (2016)
24. Hosseinzadeh, P., Bahri, O., Li, P., Boubrahimi, S.F., Hamdi, S.M.: METFORC: Classification with Meta-Learning and Multimodal Stratified Time Series Forest. In: *2023 International Conference on Machine Learning and Applications (ICMLA)*, pp. 1248–1252. IEEE (2023)
25. Karim, F., Majumdar, S., Darabi, H., Chen, S.: LSTM fully convolutional networks for time series classification. *IEEE Access* **6**, 1662–1669 (2017)
26. Ieracitano, C., Mammone, N., Hussain, A., Morabito, F.C.: A novel explainable machine learning approach for EEG-based brain-computer interface systems. *Neural Comput. Appl.* 1–14 (2021)
27. Usman, K., Rajpoot, K.: Brain tumor classification from multi-modality MRI using wavelets and machine learning. *Pattern Anal. Appl.* **20**(3), 871–881 (2017). <https://doi.org/10.1007/s10044-017-0597-8>
28. Kang, H.-W., Kang, H.-B.: Prediction of crime occurrence from multi-modal data using deep learning. *PLoS ONE* **12**(4), e0176244 (2017)
29. Yang, X., Feng, S., Wang, D., Zhang, Y.: Image-text multimodal emotion classification via multi-view attentional network. *IEEE Trans. Multimedia* **23**, 4014–4026 (2020)
30. Chen, Y., Kempton, D.J., Ahmadzadeh, A., Angryk, R.A.: Towards synthetic multivariate time series generation for flare forecasting. In: *Artificial Intelligence and Soft Computing: 20th International Conference, ICAISC 2021, Virtual Event, June 21–23, 2021, Proceedings, Part I*, vol. 20, Springer, 2021, pp. 296–307 (2021)



Fusing Image and Text Features for Scene Sentiment Analysis Using Whale-Honey Badger Optimization Algorithm (WHBOA)

Prem Shanker Yadav^{1(✉)}, Dinesh Kumar Tyagi¹,
and Santosh Kumar Vipparthi²

¹ Department of Computer Science and Engineering, Malaviya National Institute of Technology, Jaipur 302017, Rajasthan, India
{2018rcp9157,dktyagi.cse}@mnit.ac.in

² School of Artificial Intelligence and Data Engineering, Indian Institute of Technology, Ropar, Rupnagar 140001, Punjab, India
skvipparthi@iitrpr.ac.in

Abstract. Developing a real-time sentiment analysis application that relies solely on features extracted from images or textual content falls short of capturing human emotions' nuanced and multifaceted nature. The unlabeled dataset, though useful, has limitations for sentiment analysis due to its general image descriptions, which lack emotional depth and do not include direct sentiment labels. Finding scene sentiment is a challenging task. To address this, combining textual descriptions with visual features is crucial. Important parameters include entropy, bag of words, and parts of speech (nouns, adjectives, and verbs) for textual analysis, alongside visual features like SIFT, SURF, and color histograms. These features are integrated to capture a comprehensive range of sentiment cues, enhancing the accuracy and depth of sentiment insights. This paper proposes an optimized adaptive neuro-fuzzy inference system for a compelling feature enhancement using the Whale-Honey Badger Optimization Algorithm (WHBOA). The proposed method identifies the most relevant and effective features from both textual and visual data. It captures visual-specific attributes to provide a richer and more detailed representation of visual content, addressing the limitations of general image descriptions and paving the way for the development of predictive models. Additionally, text pre-processing cleans and normalizes the textual data. We conducted an extensive comparative performance evaluation to assess the effectiveness and accuracy of the proposed model. The model is compared with the Nearest Neighbor, Support Vector Machine (SVM), and Decision Tree classification algorithms for the performance assessments. The results demonstrate that the optimized model performs better, achieving an accuracy of approximately 91.2%, compared to the other models.

Keywords: Hand-crafted features · Image Captioning and Nature Inspired Optimization Algorithms

D. K. Tyagi and S. K. Vipparthi—These authors contributed equally to this work.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15302, pp. 446–462, 2025.
https://doi.org/10.1007/978-3-031-78166-7_29

1 Introduction

In contemporary times, there is a growing trend of individuals embracing social media hubs as avenues to share images and textual content and express sentiments and emotions across various events and topics. Sentiment analysis is extracting and understanding sentiments and feelings expressed in textual data. It has become progressively crucial across multiple sectors, including social media surveillance, customer sentiment analysis, and product evaluation. Traditionally, sentiment analysis has primarily focused on textual data. However, with the proliferation of multimedia content on the internet, there is a growing need to analyze sentiments in multimodal data, particularly in scenes where textual descriptions or captions accompany images. Scene sentiment analysis involves extracting sentiments and emotions from visual scenes, often accompanied by textual descriptions or captions. One of the critical challenges in this domain is the effective fusion of image and text features to capture the nuanced sentiments expressed within the scenes. Feature selection techniques play a crucial role in identifying the most discriminative features from both modalities, thereby improving the performance of sentiment analysis models. Sentiment recognition (SR) is a fundamental job in computer vision with applications ranging from human-computer interaction to affective computing and mental health assessment [1]. An effective analysis of facial expressions is crucial for understanding human emotions, intentions, and behaviors. Figure 1 visually demonstrates how the correlation between scenes and text enhances sentiment analysis compared to using only facial images. However, it is very challenging to recognize the sentiment of the view exactly. By incorporating both scene and text information, we can access the diverse features that contribute significantly to the accuracy and depth of sentiment analysis.

Face feature extraction is carried out in two ways: i) using geometric characteristics, and ii) using appearance [2]. The measurements utilized to generate the organ movement points are the source of structural geometric characteristics [3]. Using a model to analyze and predict emotions from unidentified data requires creating algorithms to identify patterns in the data already available [4]. Image-text pair multi-model sentiment recognition consists of three steps: i) feature extraction based on face tracking and identification, ii) textual features and iii) feature classification [5]. We have taken the publicly available flicker8k dataset and crafted a scene-oriented dataset by proposing a new procedure to perform the first two steps. We also investigated the proposed ANFIS-WHBO classification model to improve classification performance. Decision Tree [6], SVM [7], and k-NN [8] are models used to evaluate sentiments. The Adaptive Neuro-Fuzzy Inference System (ANFIS) is a potent computational technique that blends the human-like reasoning style of fuzzy inference systems with the learning capabilities of ANN [9]. Due to its hybrid methodology, ANFIS can handle complicated data with a high degree of generalization and precise learning capabilities. Traditional ANFIS models, however, face overfitting issues and problems with parameter optimization problems. Integrating the Whale and Honey Badger Optimization Algorithm and ANFIS resolves these problems and improves fore-



Fig. 1. a) Face-based images tell the direct expression but are not sufficient to define complete scenarios, while b) facial-based images are richer but are challenging to produce effective sentiments in real-time.

cast accuracy. Utilizing WHBOA's capacity to balance exploration and exploitation in intricate search environments enhances the ANFIS model's resilience and functionality. Traditional optimization methods may confront issues like convergence to unsatisfactory solutions or local minima. The Proposed hybrid optimization strategy, WHBOA, adopts the combined benefits of the Whale optimization Algorithm [10] (WOA) and Honey Badger Optimization Algorithm [11] (HBOA) to optimize scene sentiment analysis parameters. The WOA efficiently explores the search space inspired by humpback whale hunting behavior. In contrast, HBOA, inspired by the tenacity of honey badgers, exhibits adaptability and resilience in overcoming the optimization challenges. By integrating the face-influenced concepts with meta-heuristic optimization, our approach aims to achieve higher accuracy in sentiment analysis by effectively capturing emotional cues embedded in facial expressions [12].

Our key contributions in this paper are summarized as follows:

- Propose a hybrid algorithm for the optimized feature extraction by integrating the whale and honey badger optimization algorithms.
- We crafted faces containing scenes with their text description. This correlation allows the finding of visual and textual aspects usable in sentiment findings.
- Presented a novel framework considering the handcraft features of image-text pairs to predict the sentiment of the scene.
- Furthermore, we conducted an intensive performance analysis of the proposed model with other approaches to demonstrate its effectiveness.

We organized the rest of the paper as follows: Section 2 presents an overview of the related research on feature optimization techniques, nature-inspired optimization, and sentiment recognition. Section 3 describes the WHBOA method for sentiment recognition. Section 4 discusses the performance evaluation and results discussion. Finally, Section 5 concludes the paper with the future research directions.

2 Related Work

In recent years, there has been a surge in sentiment analysis interest in online social media, driven by the exponential growth of platforms and the influx of opinions shared by anonymous users. Siersdorfer et al. [13] pioneered the field of image sentiment analysis by proposing a machine-learning approach to predict sentiment conveyed by images on the social web, utilizing detailed pixel-level information. In [14], Rasiwasia N. et al., presented an approach to classify images as positive or negative, representing a noticeable advancement in sentiment analysis within visual content analysis. Traditionally, sentiment analysis heavily relied on lexicon-based methods and rule-based systems for classifying text sentiments [15]. Taboada et al. presented lexicon-based approaches that utilized predefined sentiment dictionaries to assign polarity scores to the words, facilitating straightforward sentiment classification [16]. Furthermore, Wilson et al. [17] presented a rule-based system, alternatively employing predefined linguistic rules to infer sentiment from textual patterns and syntactic structures. In [18], Liu et al. undertook text mining on tourist reviews, examining the perception and attention dimensions of tourists visiting Fuzhou. The authors employed topic modeling and sentiment analysis techniques such as Latent Dirichlet Allocation (LDA) and Support Vector Machines (SVM) to analyze the data. Kim et al. [19] presented an innovative method to forecast essential interest rate voting outcomes and achieved a high accuracy rate in predicting sentence sentiment using Support Vector Machines.

Islam et al. [20] considered a feature selection model coupled with word2vec and random forests to analyze the sentiment in real-time Twitter data from the 2019 election. This work is valuable for researchers and practitioners interested in understanding the current landscape and potential advancements in deep learning-based sentiment analysis. Furthermore, Jadhav et al. [21] employed a random forest (RF) machine learning algorithm and integrated demographic information for sentiment analysis. Habbat et al. proposed a multi-model combination strategy that incorporates different recurrent neural network (RNN) architectures such as Long Short-Term Memory (LSTM), Bidirectional LSTM, and Gated Recurrent Unit (GRU). They employed various word embedding techniques to enhance the performance of sentiment analysis on unstructured tweet data [22]. A soft sensor transfer approach for Long Short-Term Memory Recurrent Neural Networks (LSTM-R) presented in [23] to estimate ventilation openings by leveraging measurements of indoor and outdoor climate variables. Researchers investigated numerous strategies for combining picture and text

modalities for sentiment analysis. Jin et al. [24] introduced a multimodal sentiment analysis framework that combines picture data derived from convolutional neural networks (CNNs) with language characteristics extracted from recurrent neural networks. A graph-based fusion approach was proposed in [25] that constructs a multimodal graph to model the relationships between image regions and textual tokens for sentiment analysis.

Optimization strategies play a pivotal role in feature selection and feature weighting for sentiment analysis, minimizing classification errors and enhancing generalization. Traditional optimization methods, such as stochastic gradient descent (SGD) and its derivatives, Adam and RMSProp, are often used to improve the parameters of deep neural networks [26, 27]. However, these approaches may face difficulties such as delayed convergence or being stuck in local minima, mainly when dealing with complicated optimization landscapes [28]. Genetic Algorithms (GA) [29], Particle Swarm Optimization (PSO) [30], and Differential Evolution (DE) [31], etc., have all demonstrated success in optimizing neural network parameters, leading to notable enhancements in expression recognition performance. However, these algorithms may encounter challenges in balancing exploration and exploitation, particularly in high-dimensional and non-convex optimization scenarios. Our work provides an approach for sentiment analysis that addresses the constraints of existing optimization algorithms while also using their complementary exploration-exploitation capabilities. The proposed approach combines the properties of the WOA and HBOA to overcome the concerns mentioned above. WOA exploits the exploratory skills inspired by whale hunting behavior to efficiently explore the solution space, whereas HBOA exploits promising regions for fine-tuning. The proposed WHBOA optimization method presents a promising approach to addressing these challenges and advancing sentiment analysis research.

3 Proposed Methodology

In this section, we elaborate on the crafted database and the proposed WHOBA optimization algorithm for effective optimization of feature selection in sentiment analysis. Our optimization of the classifier model, influenced by face images within a scene, categorizes the characteristics used in the sentiment estimation system. Additionally, the performance metrics have been evaluated to assess the performance of the classifiers.

3.1 Proposed Procedure of Visual and Linguistic Features

Combining images and text in sentiment analysis creates a more impactful solution by leveraging the complementary nature of visual and textual information. By analyzing both modalities simultaneously, the model gains a deeper understanding of the sentiment expressed from the data. Figure 2 demonstrates the various steps involved in organizing the features and labels of the crafted dataset. Several visual features are used in emotion recognition. The proposed method

uses SIFT (scale-invariant feature transform), SURF (speeded-up robust features), entropy, and color histogram to identify and detect image features. NLP is applied to preprocess the text part, tokenize it, and then find features based on parameters like nouns, entropy, adjectives, verbs, and bag of words. Table 1 shows all these handcrafted visual and textual features.

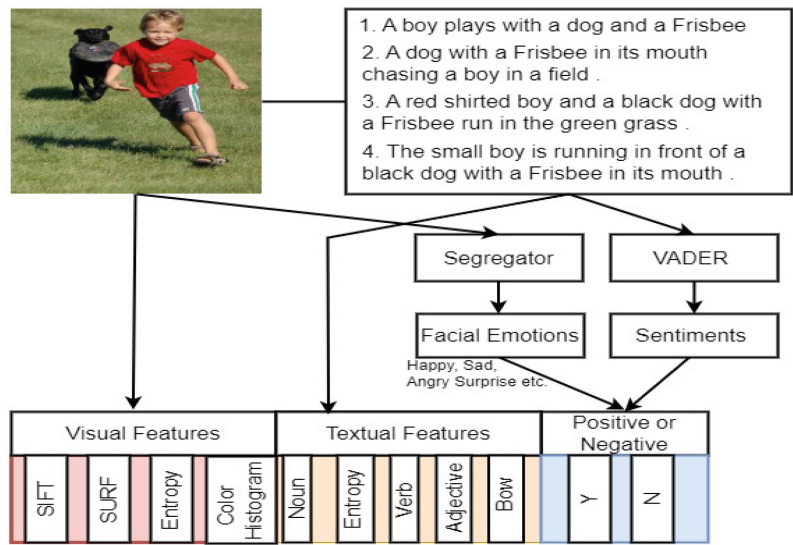


Fig. 2. Proposed procedure for crafting dataset

We used haar cascading to detect face-on images, collected 2754 face-related images and their respective captions, and applied a pre-trained face emotion detector to recognize seven emotions (happy, surprise, sad, angry, contempt, fear, and disgust), further transformed into positive and negative sentiments.

Table 1. Handcrafted Visual and Textual Features

Extracted Visual Features		Extracted Textual Features	
Entropy	Complexity or randomness of textures or patterns present in the image	Entropy	To assess the diversity of sentiments within text
Color Histogram	Frequency of occurrence of each color used to present emotional tone	Noun	Context of sentiments. Determine the subject of the text
SURF	Key points in the image based on their local intensity patterns	Adjectives	Expressing sentiment by indicating whether something is positive or negative
SIFT	Key points, scale invariance and robustness to changes in viewpoint and illumination	Verbs	Action verbs may indicate positive or negative experiences.

Text labels are classified as positive, negative, or neutral using VADER (Valence Aware Dictionary and sEntiment Reasoner), and weighted voting is applied to create the final labeling. The overall schematic framework of the proposed sentiment analysis model is shown in Fig. 3. It takes the visual and textual pairs and pre-processes the curated data. We explore the desired compelling optimal features using the WHBOA algorithm.

Finally, the sentiment label is predicted by the optimization model by integrating the handcrafted features of visual images and textual data. Each phase is depicted in Fig. 3.

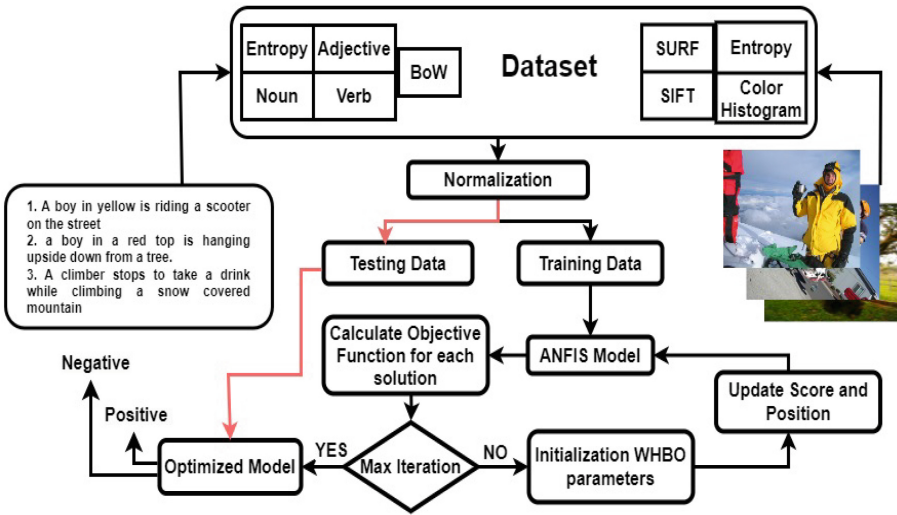


Fig. 3. A framework of the proposed model of sentiment detection enhanced by the integration of hand-crafted features of visual and textual.

3.2 Hybrid Approach of WOA and HBOA

This section presents the basics of the WOA and HBOA optimization algorithms and the proposed hybrid optimization algorithm. Optimization algorithms play a crucial role in feature selection by helping to identify the most relevant and informative subset of features from a larger pool. For feature selection, WOA and HBOA optimization approaches are promising. To enhance accuracy and identification, an integrated hybrid “Whale-Honey Badger Optimization Algorithm” (WHBOA) is proposed for an optimal subset of feature selection to improve sentiment prediction. By exploiting the behavioral orchestration of whales and honey badgers and balancing exploration with exploitation, the algorithm efficiently identifies relevant optimal features, thereby enhancing the performance of the sentiment prediction model.

3.2.1 Whale Optimization Algorithm (WOA)

The WOA algorithm [10] solves the optimization problems by simulating whales' social behavior and hunting strategies. The three actions that WOA performs include encircling prey, a bubble net attack (exploitation), and searching for prey (exploration). The fitness of each search whale is evaluated using the objective function $f_{\text{obj}}(\mathbf{X}_i)$ of each whale position \mathbf{X}_i .

$$\text{Fitness} = f_{\text{obj}}(\mathbf{X}_i) \quad (1)$$

The best solution found so far is updated accordingly. During each iteration of the WOA algorithm, the positions of the whales are updated based on the exploration and exploitation phases.

Encircling Prey

The current best candidate solution is assumed to be closest to the target prey, and other solutions update their positions toward the best search agent. The above behavior is indicated as follows:

$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D} \quad (2)$$

where, $\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a}$, $\vec{D} = |\vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}(t)|$, $\vec{C} = 2 \cdot \vec{r}$. In this 'a' decreases linearly from 2 to 0 over iterations, and \vec{r} is a random vector in the range [0, 1].

Bubble-Net Attacking(exploitation)

The humpback whales swim continuously in a spiral-shaped channel and inside a constricting circle as they move from one location to another. This shrinking and spiral behavior is mathematically represented below.

Shrinking Encircling approach:

$$\vec{X}(t+1) = \vec{X}^* - \vec{A} \cdot \vec{D} \quad (3)$$

where \vec{X}^* is the best solution found so far, and $\vec{D} = |\vec{C} \cdot \vec{X}^* - \vec{X}(t)|$.

Spiral Updating Position:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^* \quad (4)$$

where $\vec{D}' = |\vec{X}^* - \vec{X}(t)|$, b is a constant defining the shape of the logarithmic spiral, and l is a random number in the range $[-1, 1]$.

Prey Searching(exploration)

Apart from using bubble-net, humpback whales also hunt for random prey.

$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D} \quad (5)$$

$$\vec{D} = |\vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}(t)| \quad (6)$$

3.2.2 Honey Badger Optimization Algorithm (HBOA)

The HBO algorithm [11] is inspired by the adaptability and resilience of honey badgers. It involves a dynamic adjustment between the exploration and exploitation phases to solve optimization problems efficiently. The algorithm initializes the positions (X) of N honey badgers randomly within the lower and upper bounds (lb and ub). Fitness is related to the best solution of search score for each honey badger agent. It follows exploitation and exploration phases until the best solution is calculated.

Density Factor and Intensity

Density factor and intensity play roles in shaping the algorithm's behavior. Intensity is correlated with the prey's degree of concentration and the honey badger's distance from it. I_i is the prey's smell intensity; a high smell indicates a quick motion and vice versa. It is denoted as:

$$I_i = r_0 \times \frac{S}{4\pi d_i^2} \quad (7)$$

here r_0 is a random number between 0 and 1, concentration strength $S = (x_i - x_{i+1})^2$ and $d_i = x_{\text{prey}} - x_i$ denotes the directional coefficient specific to the agent.

The density factor (α) represents the concentration of potential solutions in the search space, influencing the algorithm's exploration and exploitation phases.

$$\alpha = C * \exp\left(\frac{\text{-iteration}}{\text{max_iteration}}\right) \quad (8)$$

where C is a constant.

Digging Phase

It intensifies the search around the best-known solutions, focusing on refining and exploiting promising regions of the solution space.

$$X_n = \text{Bsp} + F \cdot \beta \cdot I \cdot \text{Bsp} + F \cdot r1 \cdot \alpha \cdot (d) \cdot (\cos(2\pi r2) \cdot (1 - \cos(2\pi r3))) \quad (9)$$

A new candidate solution X_n is generated by perturbing a base point Bsp using random or directional components $r1$, $r2$ and $r3$, α , d , and scaling factor F .

Honey Phase

It Diversifies the search, potentially exploring new and less-explored regions of the solution space to discover better solutions. This perturbation helps diversify the search and explore potentially better solutions in the solution space. Here $r4$ is a random number between 0 to 1.

$$X_n = \text{Bsp} + F \cdot r4 \cdot \alpha \cdot d; \quad (10)$$

3.2.3 Proposed Whale-Honey Badger Optimization Algorithm

The WHBOA combines the principle behaviors of WOA and HBO for enhanced performance in solving the feature selection as an optimization problem described in Algorithm 1. Integrating the Whale and Honey Badger Optimization strategies in WHBOA facilitates robust sentiment analysis by dynamically adjusting between global exploration and local exploitation. This balance enhances the algorithm's ability to converge towards optimal sentiment scores over the course of iterations. The steps involved in the algorithm are described as follows:

Initialization

The algorithm begins by initializing the positions (Agent_Positions) of N search agents randomly within the specified bounds (lb and ub).

Fitness Function

The fitness function, f_{obj} , evaluates the fitness of each search agent based on its position.

$$\text{Fitness} = f_{obj}(\mathbf{X}_j) \quad (11)$$

The purpose is to maximize the fitness function, i.e., classification accuracy metric.

Algorithm 1. WHBOA Algorithm for feature selection

- 1: **Input:** (N , Max.Iteration, lb, ub, dim, fobj, Dataset with features)
 - 2: **Initialization:** Initialize positions of search agents randomly within bounds (lb, ub), Representation of features set
 - 3: **Output:** Optimal solution (Agent Best score), best features
 - 4: **Begin**
 - 5: Calculate and Evaluate Fitness of search
 - 6: **for** Iter = 1 to Max.Iteration **do**
 - 7: WHBOA Fitness evaluation
 - 8: Calculate intensity and density factor using Eq. (7) & Eq. (8)
 - 9: Evaluate Agent_score = fobj(Agent_Positions)
 - 10: **for** j = 1 to dim **do**
 - 11: Calculate distance and update positions using Eq. (12) & Eq. (13)
 - 12: Calculate direction and update positions using Eq. (14)
 - 13: **end for**
 - 14: **end for**
 - 15: **end**
 - 16: **Return:** Set of Selected features
-

Intensity and Density Factor

The density factor, α , influences the selection and movement of solutions within the search space. It dynamically adjusts exploitation and exploration phases based on the current iteration. Equation (7) and Eq. (8) are used to calculate intensity and density factor.

Exploitation

During periods of high density (indicative of local convergence), the algorithm exploits known solutions to refine them further. This phase involves intensively exploring the immediate vicinity of promising solutions to potentially converge toward the global optimum.

$$\vec{DB} = \left| \vec{C} \cdot \vec{BP} - \vec{AP} \right| \quad (12)$$

$$\vec{AP} = \vec{BP} - \vec{A} \cdot \vec{DB} \quad (13)$$

where, \vec{C} and \vec{A} are specified in Eq. (2).

Exploration

During the exploration, agents broaden their scope to explore new regions of the solution space, inspired by curiosity and randomness.

$$\vec{AP} = \vec{BP} + F \cdot r4 \cdot \alpha \cdot I \cdot k \cdot \vec{DB} + F \cdot r1 \cdot \alpha \cdot (1 - k) \cdot \vec{DB} \cdot |\cos(2\pi r2) \cdot (1 - \cos(2\pi r3))| \quad (14)$$

where, k is 0 or 1.

3.3 Adaptive Neuro-Fuzzy Inference System-ANFIS

It is a hybrid intelligent system that combines the interpretability of fuzzy logic systems with the adaptive capabilities of artificial neural networks (ANNs) [12]. By learning from data and producing language rules, it seeks to represent complicated interactions between inputs and outputs. This makes it appropriate for situations where it is challenging to express explicit rules or when the interpretability of the model's conclusions is crucial. ANFIS comprises nodes and routed pathways as shown in Fig. 4, and all input-output values are modifiable by changing the network design's parameters. This ANFIS is used to evaluate the efficacy of sentiment prediction through the extracted features using the WHBOA algorithm.

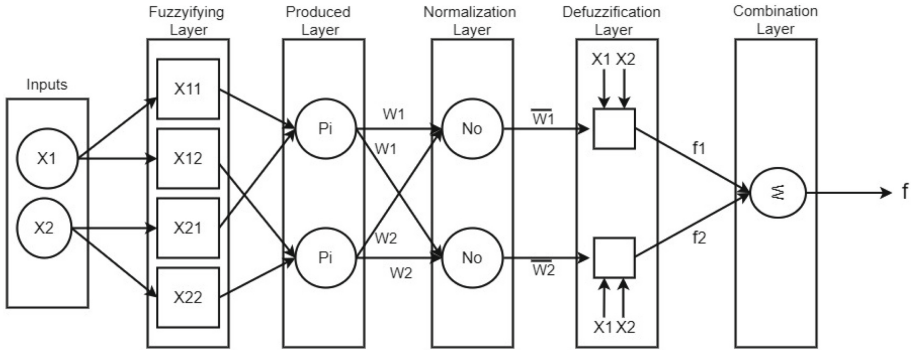


Fig. 4. ANFIS structure

4 Experiment and Evaluation

A link between many variables can be established, and a related variable can be predicted based on one or more independent elements using different methodologies such as Linear regression, neural networks, fuzzy inference systems, and a mixture of fuzzy inference systems and neural networks. Based on the hybrid optimization algorithm and adaptive neuro-fuzzy inference system (ANFIS) for prediction, the performance of the combined technique is investigated in this paper. The method uses various machine learning algorithms to investigate whether the scene sentiment is properly classified.

4.1 Dataset

We evaluated our work on the dataset crafted from flicker8k. This is unlabeled data for sentiment analysis but gives a long range of general scenes or views. Flickr8k comprises a collection of images gathered from the popular photo-sharing platform Flickr8k, along with human-annotated descriptions for each image. The dataset contains approximately 8,000 images, each paired with five unique captions, resulting in a total of around 40,000 caption-image pairs. For analysis, the data is prepared and cleaned. The handcrafted method extracts features meticulously tailored from visual and textual data to ensure that it effectively captures nuanced aspects. We constructed a robust representation poised to facilitate accurate sentiment analysis across varied data modalities using distinct features extracted from both the visual and textual data.

4.2 Performance Evaluation

In our experiments, the datasets are divided into training and testing sets, using a split ratio of 7:3. Our framework is implemented using MATLAB. We considered widely used metrics- accuracy, precision, recall, and F1-score to evaluate the efficacy of the proposed work.

$$\begin{aligned} \text{Precision} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \\ \text{Recall} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \\ \text{F1-score} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

4.3 Results

This section evaluates and compares the performance of the WHBOA-based ANFIS classifier with Nearest Neighbor, SVM, and Decision Tree classifiers. Accuracy, precision, recall, and F1-score statistics in sentiment recognition are used to systematically assess the efficacy of the proposed method. The performance results are shown in Table 2.

Table 2. ANFIS Performance Comparison with KNN, SVM, and DT

Method	Accuracy (%)	Precision	Recall	F1 Score
Nearest Neighbor Classifier	88.50	90.02	90.58	90.29
Support Vector Machines	90.57	90.39	90.92	90.85
Decision Tree	89.32	90.17	90.74	90.45
WHBOA-based ANFIS	91.21	90.62	91.28	90.94

The results show that our optimization approach outperformed compared to the considered baseline models. Table 3 shows the comparison outcome with the existing optimization algorithms, the Lion Optimization Algorithm (LOA), Whale Optimization Algorithm (WOA), Honey Badger Optimization Algorithm (HBOA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO).

Table 3. Performance with other existing Optimization Algorithms

	LOA	WOA	HBOA	PSO	ACO	WHBOA
Accuracy	90.3766	90.7950	90.3766	90.3766	90.7950	91.2134
Precision	89.7626	90.3911	89.7151	89.9038	90.5141	90.6223
Recall	90.4020	90.3911	90.7602	90.0439	90.2120	91.2756
F-Score	90.0812	90.3911	90.2346	89.9738	90.3628	90.9478
Error	9.6234	9.2050	9.6234	9.6234	9.2050	8.7866

The efficacy of the method is evident and can be clearly visualized in Fig. 5. The results show that this approach outperforms existing models in sentiment classification.

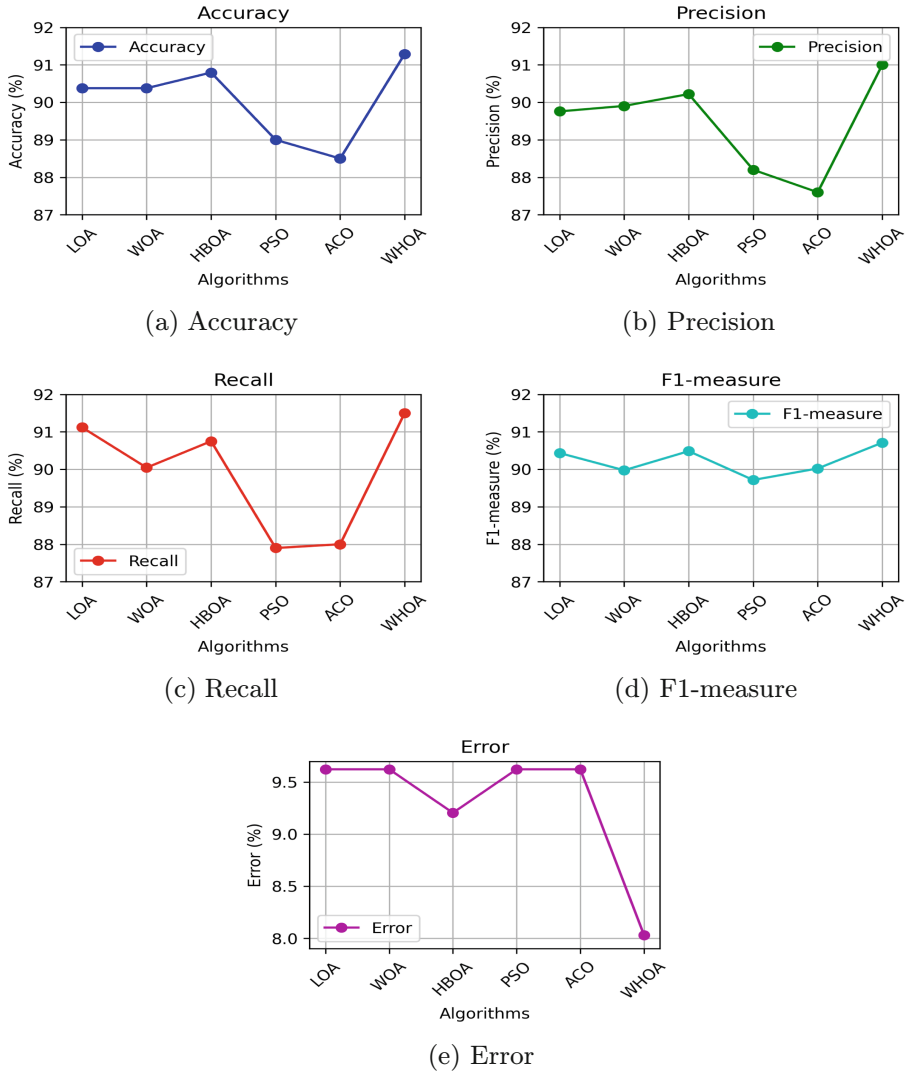


Fig. 5. Comparative results of a) accuracy, b) precision, c) recall, d) F1 score and e) error.

4.4 Ablation Study

We have designed and executed ablation studies using various combinations, including KNN with LOA, WOA, HBOA, PSO, ACO, and WHBOA, as well as SVM, DT, and ANFIS with LOA, WOA, HBOA, PSO, ACO, and WHBOA optimizers. Our findings indicate that the classifiers KNN, SVM, DT, and ANFIS when optimized with WHBOA, achieve the highest performance. Therefore, we have included only the results with WHBOA optimization in Table 2.

5 Conclusion and Future Work

This paper presents WHBOA, a hybrid optimization algorithm inspired by whales and honey badgers, enhancing sentiment analysis by extracting key visual and textual features from datasets. This method helps to improve the effectiveness of the sentiment finding in sentiment analysis. We developed a robust dataset of face-influenced scene images along with their text descriptions, annotated with positive or negative sentiments, to rigorously test our approach. We evaluated the performance efficiency of the WHBOA algorithm with the ANFIS classifier. The results demonstrate that this method outperforms Nearest Neighbor, SVM, and Decision Tree in terms of accuracy, precision, recall, and F1-score. The proposed WHBOA-based ANFIS classifier achieves an accuracy of 91.21%. It also shows better performance efficacy than the LOA, WOA, HBOA, PSO, and ACO meta-heuristic optimization algorithms. Future work will address facial occlusions, image region alignment, and semi-supervised approaches in image retrieval.

Acknowledgement. This work was made possible through the generous support of the project SERB/2022/000332. The authors would also like to thank all the members of the CVPR and Vision Intelligence Lab for their support in this work.

Declarations

Funding. This research received no specific funding.

Conflict of interest. The authors declare that they have no competing interests.

Ethics approval. The research conducted for this manuscript did not require ethics approval as it did not involve human participants or animals.

Authors' contributions. All authors significantly contributed to the design, manuscript revision, and final approval.

References

1. Ekman, P.: Facial expression and emotion. *Am. psychol.* **48**, 384 (1993)
2. Dornaiika, F., Moujahid, A., Raducanu, B.: Facial expression recognition using tracked facial actions: classifier performance analysis. *Eng. Appl. Artif. Intell.* **26**, 467–477 (2013)
3. Loconsole, C., Miranda, C.R., Augusto, G., Frisoli, A., Orvalho, V.: Realtime emotion recognition novel method for geometrical facial features extraction, Vol. 1, pp. 378–385. *IEEE* (2014)
4. Ahuja, R., Chug, A., Kohli, S., Gupta, S., Ahuja, P.: The impact of features extraction on the sentiment analysis. *Procedia Comput. Sci.* **152**, 341–348 (2019) extraction on the sentiment analysis. *Procedia Comput. Sci.* **152**, 341–348 (2019)
5. Zhu, T., et al.: Multimodal sentiment analysis with image-text interaction network. *IEEE Trans. Multimedia* **25**, 3375–3385 (2022)

6. Bilal, M., Israr, H., Shahid, M., Khan, A.: Sentiment classification of roman-urdu opinions using naïve bayesian, decision tree and knn classification techniques. *J. King Saud Univ.-Comput. Inf. Sci.* **28**, 330–344 (2016)
7. Chandra, M.A., Bedi, S.: Survey on svm and their application in image classification. *Int. J. Inf. Technol.* **13**, 1–11 (2021)
8. Guo, X.: A KNN classifier for face recognition. In: 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), pp. 292–297. IEEE (2021)
9. Iqbal, M., Raza, S.A., Abid, M., Majeed, F., Hussain, A.A.: Artificial neural network based emotion classification and recognition from speech. *Int. J. Adv. Comput. Sci. Appl.* **11** (2020)
10. Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016)
11. Hashim, F.A., Houssein, E.H., Hussain, K., Mabrouk, M.S., Al-Atabany, W.: Honey badger algorithm: new metaheuristic algorithm for solving optimization problems. *Math. Comput. Simul.* **192**, 84–110 (2022)
12. Dirik, M.: Optimized anfis model with hybrid metaheuristic algorithms for facial emotion recognition. *Int. J. Fuzzy Syst.* **25**, 485–496 (2023)
13. Siersdorfer, S., Minack, E., Deng, F., Hare, J.: Analyzing and predicting sentiment of images on the social web. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 715–718 (2010)
14. You, Q., Luo, J., Jin, H., Yang, J.: Robust image sentiment analysis using progressively trained and domain transferred deep networks, vol. 29 (2015)
15. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. arXiv preprint [arxiv: cs/0205070](https://arxiv.org/abs/cs/0205070) (2002)
16. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **37**, 267–307 (2011)
17. Xi, R.: A comprehensive review of text sentiment analysis: a survey of traditional methods and deep learning approaches. *Sci.Technol. Eng. Chem. Environ. Prot.* **1**(5) (2024)
18. Liu, R., Lin, J., Wei, Q., Jiang, Q.: Fuzhou destination image perception study: based on machine learning lda model and svm model, vol. 12604, 973–979 (SPIE, 2023)
19. Kim, M., Cho, S.: Monetary policy document analysis for prediction of monetary policy board decision. *Heliyon* **9**(10) (2023)
20. Hitesh, M., Vaibhav, V., Kalki, Y.A., Kamtam, S.H., Kumari, S.: Real-time sentiment analysis of 2019 election tweets using word2vec and random forest model, pp. 146–151. IEEE (2019)
21. Jadhav, P., et al.: Application of a machine learning algorithm in prediction of abusive head trauma in children. *J. Pediatr. Surg.* **59**, 80–85 (2024)
22. Habbat, N., Anoun, H., Hassouni, L.: Combination of GRU and CNN deep learning models for sentiment analysis on French customer reviews using XLNet model. *IEEE Eng. Manage. Rev.* **51**, 41–51 (2022)
23. Guesbaya, M., García-Mañas, F., Rodríguez, F., Megherbi, H.: A soft sensor to estimate the opening of greenhouse vents based on an LSTM-RNN neural network. *Sensors* **23**, 1250 (2023)
24. Jin, Z., Cao, J., Guo, H., Zhang, Y., Luo, J.: Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In: Proceedings of the 25th ACM International Conference on Multimedia 795–816 (2017)
25. Tan, Q., Shen, X., Bai, Z., Sun, Y.: Cross-modality fused graph Convolutional network for image-text sentiment analysis, pp. 397–411. Springer (2023)

26. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
27. Dauphin, Y.N., et al.: Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Adv. Neural Inf. Process. Syst.* **27** (2014)
28. Sun, R.-Y.: Optimization for deep learning: an overview. *J. Operat. Res. Soc. China* **8**, 249–294 (2020) *Operat. Res. Soc. China* **8**, 249–294 (2020) *Operations Research Society of China* **8**, 249–294 (2020)
29. Mirjalili, S., Mirjalili, S.: Genetic algorithm. *Evol. algorithms neural netw.: theory appl.* 43–55 (2019)
30. Kennedy, J., Eberhart, R.: Particle swarm optimization, Vol. 4, pp. 1942–1948. IEEE (1995)
31. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**, 341–359 (1997)



EncodeNet: A Framework for Boosting DNN Accuracy with Entropy-Driven Generalized Converting Autoencoder

Hasanul Mahmud^(✉), Palden Lama, Kevin Desai, and Sushil K. Prasad

Department of Computer Science, The University of Texas at San Antonio,
San Antonio, TX 78249, USA
hasanul.mahmud@my.utsa.edu,
{palden.lama, kevin.desai, sushil.prasad}@utsa.edu

Abstract. Image classification is a fundamental task in computer vision, and the quest to enhance DNN accuracy without inflating model size or latency remains a pressing concern. We make a couple of advances in this regard, leading to a novel EncodeNet design and training framework. The first advancement involves Converting Autoencoders, a novel approach that transforms images into an easy-to-classify image of its class. Our prior work that applied the Converting Autoencoder and a simple classifier in tandem achieved moderate accuracy over simple datasets, such as MNIST and FMNIST. However, on more complex datasets like CIFAR-10, the Converting Autoencoder has a large reconstruction loss, making it unsuitable for enhancing DNN accuracy. To address these limitations, we generalize the design of Converting Autoencoders by leveraging a larger class of DNNs, those with architectures comprising feature extraction layers followed by classification layers. We incorporate a generalized algorithmic design of the Converting Autoencoder and intraclass clustering to identify representative images, leading to optimized image feature learning. Next, we demonstrate the effectiveness of our EncodeNet design and training framework, improving the accuracy of well-trained baseline DNNs while maintaining the overall model size. EncodeNet's building blocks comprise the trained encoder from our Generalized Converting Autoencoders transferring knowledge to a lightweight classifier network - also extracted from the baseline DNN. Our experimental results demonstrate that EncodeNet improves the accuracy of VGG16 from 92.64% to 94.05% on CIFAR-10, and ResNet20 from 74.56% to 76.04% on CIFAR-100. It outperforms state-of-the-art techniques that rely on knowledge distillation and attention mechanisms, delivering higher accuracy for models of comparable size.

Keywords: Converting Autoencoder · Entropy · Clustering · Deep Neural Networks

P. Lama, K. Desai and S.K. Prasad—Contributing authors.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15302, pp. 463–477, 2025.
https://doi.org/10.1007/978-3-031-78166-7_30

1 Introduction

In recent years, deep learning models have gained tremendous success in many computer vision tasks such as image classification [1, 2], object detection [3, 4], and semantic segmentation [5, 6]. However, this also brings the challenge of creating models that strike a balance between accuracy and efficiency. Although larger DNN models exhibit high accuracy in computer vision tasks, they are computationally expensive. Therefore, there is a growing interest in designing efficient DNNs to achieve high accuracy while maintaining low computational cost. One of the approaches to achieve this goal is model compression, which involves pruning and quantization of large models while preserving their accuracy to some extent [7–13]. Alternatively, techniques like knowledge distillation [14–17], and attention mechanism [18–20] focus on improving the accuracy of baseline models with either minimal or no increase in the model size. Our work is in line with the latter approach.

In our prior work [21], we introduced a Converting Autoencoder (CAE) that transforms input images into easy-to-classify images of its class and subsequently processed the transformed images with a simple classifier. The CAE model, which was based on the U-Net architecture relied on early-exiting DNN framework for its training and achieved moderate accuracy over simple datasets such as MNIST, FMNIST, etc. However, its accuracy on more complex datasets like CIFAR-10 was limited to 78–80% when passed to AlexNet or ResNet50 for inference. In this paper, we generalize the design of our Converting Autoencoder by systematically deriving its structure from given baseline DNN, making it applicable to a large class of DNNs and more complex datasets. Leveraging the Generalized Converting Autoencoder (GCAE), we developed *EncodeNet*, a novel integrative framework that enhances the accuracy of any baseline DNN with a modular architecture of feature extraction layers followed by classification layers, achieving performance on par with significantly larger models. Our framework surpasses competing techniques, including state-of-the-art knowledge distillation and attention mechanism-based methods.

EncodeNet involves two stages of model training. In the initial stage, a converting autoencoder is trained to transform an image into a representative image within the same class, thus extracting its salient features. To identify the most representative images, each class of images is grouped into different clusters based on their similarity. For each cluster, the image that can be classified with the lowest entropy using the baseline DNN is selected as the representative image. A low entropy of a classification result indicates a high confidence in the prediction. Intraclass clustering effectively reduces the reconstruction loss of the converting autoencoder. In the second stage, we combine the encoder layers of the trained converting autoencoder with additional layers and filters derived from the classification layers of the baseline DNN model. We train the new DNN by freezing the pre-trained encoder layers and only training the remaining layers of the network. By doing so, we leverage the learned representations from the autoencoder and fine-tune them for image classification. To our knowledge, this is the first integrative framework designed for entropy-driven representative feature extrac-

tion with the help of a Generalized Converting Autoencoder. It synergizes its capabilities with a thin subnetwork extracted from a baseline DNN, yielding an equivalent DNN with significantly improved accuracy.

Our key contributions are as follows:

- We developed an algorithmic approach to generalize the design of our Converting Autoencoder from U-Net architecture and early-exiting DNNs to a larger class of DNNs and more complex datasets than previously possible.
- We designed *EncodeNet*, a new framework leveraging Generalized Converting Autoencoder for training lightweight DNNs that can achieve accuracy comparable to significantly larger models without increasing model size. Our approach competes well with state-of-the-art techniques, including Knowledge Distillation and Attention Mechanism, due to its versatility. It can be implemented even when large teacher models are unavailable, yet it still attains comparable or greater accuracy.
- Experimental results using CIFAR-10 and CIFAR-100 datasets demonstrate the remarkable effectiveness of our approach, outperforming competing techniques. EncodeNet improves the accuracy of VGG16 from 92.64% to 94.05% on CIFAR-10, and ResNet20 from 74.56% to 76.04% on CIFAR-100. It outperforms KD (Knowledge Distillation) [14], RKD (Relational Knowledge Distillation) [22], FitNet [17], and FT (Factor Transfer) [16] for both ResNet and VGG networks. EncodeNet enhances the accuracy of ResNet50 on CIFAR-100 from 77.23% to 80.1%, outperforming attention mechanisms based techniques, Squeeze-and-Excitation Networks (SE) [23] and Bottleneck Attention Module (BAM) [18]. It achieves comparable accuracy with Convolutional block attention module (CBAM) [19], and Global attention mechanism (GAM) [20] while maintaining relatively small model size.

2 Related Work

In this section, we review some of the significant methods that focus on improving DNN accuracy with either minimal or no increase in the model size.

2.1 Autoencoders

An autoencoder is an artificial neural network that learns efficient encodings of unlabeled data. It consists of an Encoder, which learns how to encode data into a reduced representation efficiently, and a Decoder, which learns how to reconstruct the data back to a representation that is as close to the original input as possible [24]. There are several variations of autoencoders. The denoising autoencoders are trained to recover original input from intentionally perturbed or noisy input [25], with the aim to learn a more robust representation of input data. A variational autoencoder is a generative model that can produce different variations of existing data samples [26]. Converting autoencoders [27] has been used

to improve the performance of the early-exiting framework. However, this approach is limited to simple networks and datasets. In addition, it relies on early-exiting frameworks, further limiting its capability. Autoencoders are also widely used for dimensionality reduction, denoising, data augmentation, and anomaly detection. What sets us apart is our synergistic system approach integrating an autoencoder trained for representative image transformation, transferring its knowledge to a thin subnetwork of a baseline network yielding lightweight DNNs for resource-constrained devices.

2.2 Knowledge Distillation

Initially proposed by Hinton et al. [14], knowledge distillation is a model compression method that distills the knowledge in a large network or ensemble of networks into a small student network by forcing the student's predictions to match those of the teacher. The effectiveness of distillation has been demonstrated in many studies, e.g. [28–30], using various student and teacher architecture patterns with different depths and widths. More recently, Beyer et al. [15] identified that aggressive data augmentation and a long training schedule can drastically improve the effectiveness of the original knowledge distillation method proposed in [14]. Our framework presents an alternative method that results in a lightweight model, ideal for devices with limited resources. It does not depend on large models, yet it surpasses state-of-the-art methods based on knowledge distillation.

2.3 Streamlined DNN Architectures

Several works have proposed methods to streamline DNN architectures for resource-constrained platforms. Adadeep [31] is a usage-driven, automated DNN compression framework that systematically explores the trade-off between performance and resource constraints. SubFlow [32] uses subnetwork pruning to find the optimal subnetworks for each layer that can preserve the accuracy of the original network, and fine-tunes them with a sparsity regularization term. MobileNet [33] and MobileNetV2 [34] are DNN architectures designed for mobile and embedded vision applications, which, respectively, use depth-wise separable convolutions and inverted residual blocks with bottle-necking features to reduce the number of parameters and computations. MobileNetV2 also introduces linear bottlenecks and shortcut connections to improve the model's efficiency and accuracy. Unlike these approaches, our work focuses on enhancing the accuracy of existing DNN models without introducing new network architectures.

2.4 Attention Mechanisms

Several studies have focused on performance improvements using attention mechanisms for image classification tasks. The attention mechanism is a computational technique enabling neural networks to concentrate on pertinent input

data segments during task execution. This mirrors the selective attention mechanism observed in human cognition, allowing models to assign varying levels of significance to distinct segments of the input sequence. Squeeze-and-Excitation Networks (SENet) [23] is the first to use channel attention and channel-wise-feature-fusion to suppress the unimportant channels. The convolutional block attention module (CBAM) [19] places the channel and spatial attention operation sequentially, while the bottleneck attention module (BAM) [18] did it in parallel. Global attention [20] mechanism was proposed that boosts the performance of deep neural networks by reducing information reduction. Even though attention mechanisms have been successful in increasing DNN performance, additional parameters are needed to learn how to weigh the importance of different parts of the input sequences.

3 Method

In this section, we introduce the details of our proposed EncodeNet framework, which comprises of three key aspects - (1) generalizing the Converting Autoencoders for efficient training setup, (2) intraclass clustering and entropy-based image selection to support representative feature learning with Converting Autoencoder, and (3) knowledge transfer from Converting Autoencoder for image classification. Figure 1 provides the details of the framework, with more details provided in the sections below.

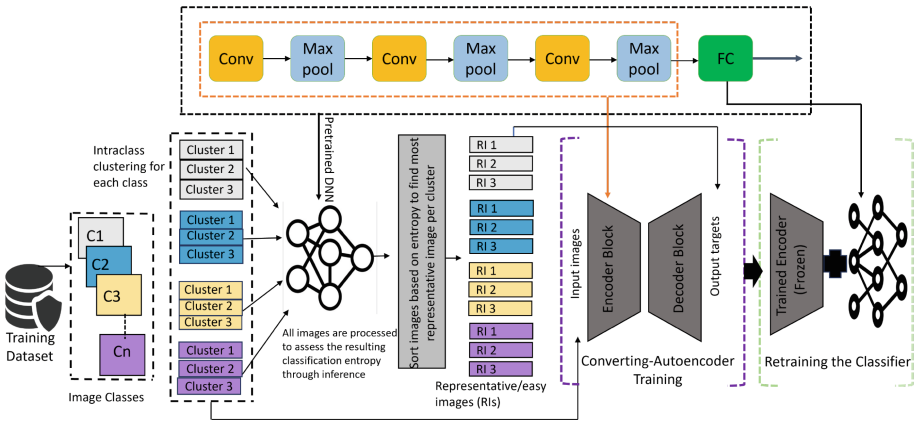


Fig. 1. Overview of the EncodeNet Framework: Intraclass clustering of images, followed by sorting them according to their classification entropy to pinpoint the most representative image for each cluster. Subsequently, a converting autoencoder is trained to transform input images into their corresponding representative images within the same class and cluster. Finally, the trained encoder layers are detached from the autoencoder, coupled with fully connected layers, and retrained to perform classification.

3.1 Converting Autoencoder Design

Autoencoders are widely employed for various tasks such as image reconstruction, image compression, denoising, dimensionality reduction, etc. In our attempt, we crafted an autoencoder to transform images into easily classifiable representations within their respective classes. Designing a lightweight autoencoder poses challenges, particularly when the encoder component can not successfully extract features, potentially leading to miss-transformations. While the UNet [35] model offers a built-in architecture for encoder and decoder blocks, it fails to convert images into easy representations. Thus, our primary objective was developing an encoder capable of accurately capturing features and designing the decoder block for efficient transformation. In general, DNN architecture for the image classification task consisted mainly of feature extraction and classification layers. The DNN's feature extraction stage plays a crucial role in capturing vital features and reducing the size of feature maps. The classification layers utilize these identified features to make predictions. Considering this, our framework commences by selecting a baseline for autoencoder design using the feature extraction layers as the encoder of the autoencoder, as depicted in Figure 1. Specifically, our focus is on the feature extractor component of the Deep Neural Network (DNN), with the exclusion of classification layers. Next, we create the corresponding decoder block, a meticulous process aimed at complementing the feature extraction mechanism and efficiently reconstructing the input data. Various architectures were employed to optimize the decoder block, aiming to reduce the autoencoder's reconstruction loss. Convolutional layers and Upsampling layers were utilized to upscale the extracted feature maps to their original dimensions. By discarding classification layers in this phase, our objective is to design a customized autoencoder capable of capturing and representing crucial features from the input data, thereby paving the way for subsequent improvements in model performance.

3.2 Representative Feature Learning with Converting Autoencoder

3.2.1 Intraclass Clustering

Complex datasets such as CIFAR-10, CIFAR-100 can have high dissimilarity among images of the same class, which poses a challenge for representative image transformation with Converting Autoencoders. For example, Fig. 2 shows that the airplane class in CIFAR-10 dataset contains various types of airplanes (e.g., fighter jet, commercial airplane, etc.) that differ significantly from each other. It is more difficult and costly to transform a fighter jet into a representative image of a commercial airplane than into a representative image of a fighter jet. To tackle this problem, we introduce intraclass clustering that groups images of the same class into different clusters based on their similarity. This facilitates effective training for our Converting Autoencoder to perform the representative image transformation within each cluster. Table 1 shows the effectiveness of intraclass clustering in minimizing the reconstruction loss for the conversion of the images.

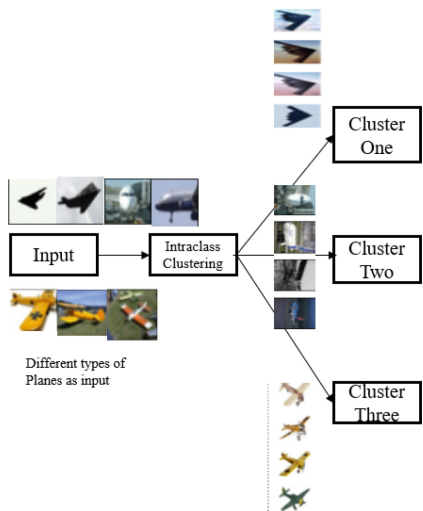


Fig. 2. Intraclass clustering on airplane class of CIFAR-10.

Table 1. Impact of intraclass clustering in minimizing the reconstruction loss on CIFAR-10 datasets after 500 epochs of training

Reconstruction loss of converting autoencoder		
Models	Without Clustering	With Clustering
VGG8	0.019	0.008
VGG16	0.013	0.07
ResNet18	0.016	0.009
ResNet20	0.025	0.0107

For efficient clustering, we extracted important features of the input image using a VGG-16 [36] model trained on ImageNet [37] data. The pre-trained VGG16 model is a convolutional neural network (CNN) model that has already been trained on a large dataset, typically for image recognition tasks. In this case, the input image to the VGG16 model had a shape of (32,32,3), which means it had a width and height of 32 pixels and 3 color channels (red, green, and blue). When using VGG16 for feature extraction, it's common to remove the last few layers of the network, including the fully connected and classification layers, and use the remaining layers as a feature extractor. We fed input images to VGG16 and used the 13th layer to extract a feature map with a shape of (1,1,512). This reduced shape makes it more suitable for applying a clustering algorithm, which aims to group similar data points together based on their features. We applied three different clustering techniques to group similar images together within each class: k-means clustering, HDBSCAN, and cosine similarity clustering. Among these three techniques, k-means clustering provided the best results.

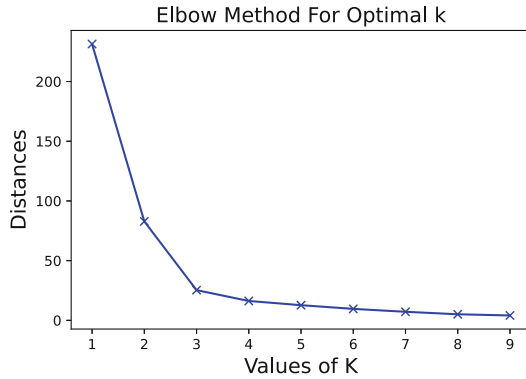


Fig. 3. Finding the optimal number of clusters.

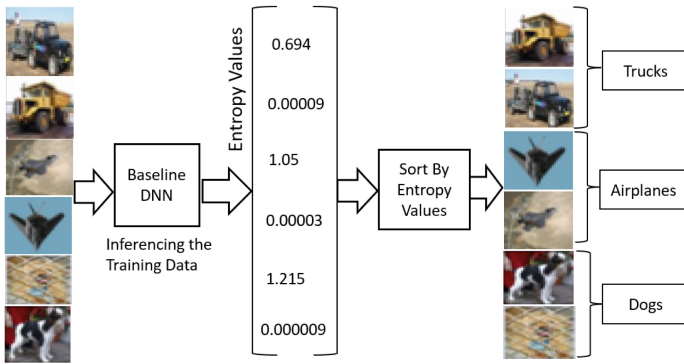


Fig. 4. Measuring the entropy of baseline DNN model.

To determine the optimal number of clusters for k-means clustering, we used the Elbow method [38], which is a technique that plots the sum of squared distances between data points and their assigned cluster centers for different values of k (the number of clusters). The “elbow” point on this plot represents the value of k where the decrease in sum of squared distances begins to level off. As shown in Fig. 3, we found that the elbow point was at k=3. We aimed to group similar images together into 3 distinct clusters within each class of the CIFAR10 dataset.

3.2.2 Entropy-Based Representative Image Extraction

The level of difficulty in the image-classification task and the requirement for complex DNN models depend on the complexity of the input. We measure the confidence of a classification result produced by a baseline DNN model (e.g., VGG, ResNet) based on the entropy of the prediction. If an input image is pre-

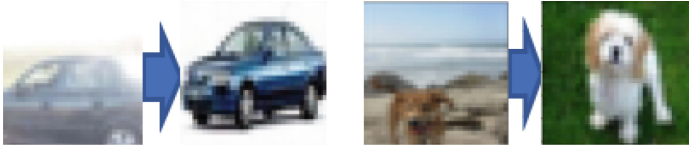


Fig. 5. A converting autoencoder transforming CIFAR10 images into the representative images of their corresponding class and cluster.

dicted to belong to a single class with 100% probability, the entropy is zero, as there is no uncertainty in the classification. If the prediction probability is equally distributed across all classes, the entropy is at its maximum. As shown in Fig. 4, we first sort the images in each cluster and class by their entropy. The representative image is identified as the one that has the lowest entropy in the corresponding class and cluster. The next stage of the framework trains a Converting Autoencoder to transform any input image of a particular class and cluster into the corresponding representative image.

3.2.3 Representative Feature Learning Through Image Transformation

We train our Converting Autoencoder to encode any given input image into an efficient representation that can be decoded into the representative image belonging to the same class and cluster. We evaluated the effectiveness of our approach using the CIFAR-10 and CIFAR-100 datasets with and without intraclass clustering. The Converting Autoencoder’s performance was evaluated in terms of reconstruction loss, which measures the similarity between the output and the target image. We set the mse (mean squared error) as the loss function of the autoencoder. The target image is the easiest image within the cluster for a given input image belonging to a cluster. As a result of the intraclass clustering, the reconstruction loss of our autoencoder dropped by 46%, from 0.0012 to 0.0065. Figure 5 illustrates the transformation of the given images to the corresponding representative images using the autoencoder trained with CIFAR-10 dataset.

3.3 Knowledge Transfer from Converting Autoencoder for Image Classification

In the EncodeNet framework, the encoder layers of an autoencoder captures the essential features of the given input image and the representative image that can aid in efficient image classification. Therefore, we employ transfer learning to train a DNN model that consists of the pre-trained encoder layers and additional layers obtained through the classification part of a baseline DNN model. By freezing the pre-trained encoder layers and only training the remaining layers of the network, we utilize the learned representations from the Converting

Autoencoder and adapt them for the task of image classification. This approach integrates the Converting Autoencoder’s representative image transformation capability into an efficient DNN model with enhanced accuracy.

4 Experiments and Results

In this section, we discuss the experiments we conducted on the different datasets to evaluate our proposed EncodeNet framework against other competing techniques. We first provide the implementation details of our work and present the datasets we have used in our experiments. Next, we provide results demonstrating the enhanced performance achieved for the baseline DNNs as a result of our EncodeNet framework. We also conduct ablation studies to evaluate the importance of our proposed use of intraclass clustering and the auto-encoder framework. Lastly, we compare our approach against widely used methods for Knowledge Distillation and Attention mechanisms that aim at enhancing the performance of the given DNN model.

4.1 Implementation Details

Our model is built on the TensorFlow framework [39] and trained on two datasets: CIFAR-10 and CIFAR-100 as described in 4.2. For the sake of comparison with Knowledge Distillation frameworks, we utilized the repository [40] that was built on the PyTorch framework [41]. For training the baseline DNN models, we set the learning rate $1e-1$ and a weight decay of $1e-4$. We conducted training sessions for the baseline models over 300 epochs. For the autoencoder aimed at converting images into simpler representations, an extended training period of 500 epochs was necessary to minimize loss effectively. The optimization objective for the Converting Autoencoders centered on minimizing the reconstruction loss measured in terms of mean squared error (MSE).

4.2 Datasets

We evaluated our proposed framework on two widely used image classification datasets: CIFAR-10 [42], CIFAR-100 [43]. The CIFAR-10 dataset contains 60,000 color images of size 32×32 , of which 50,000 are used for training and 10,000 for testing. The dataset includes 10 different classes of images, consisting of airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Whereas the CIFAR-100 dataset has 100 classes containing 600 images each, of which 500 are used for training and 100 for testing.

4.3 Improvement of Baseline DNN

Table 2 presents results illustrating the improvement in accuracy for the baseline network achieved through the robust training of our framework. For comparison, we evaluated various derivatives of two types of networks: VGG [36] and

Table 2. Results of our EncodeNet framework used to improve the accuracy (in %) of different baseline DNN models on the CIFAR-10 and CIFAR-100 datasets.

Architectures	CIFAR-10		CIFAR-100	
	Baseline	EncodeNet	Baseline	EncodeNet
VGG8	89.25	91.60	70.23	73.41
VGG16	92.64	94.06	73.37	75.11
ResNet18	91.12	92.87	74.22	75.29
ResNet20	92.56	93.64	74.56	76.04

ResNet [1]. For example, VGG8 exhibited an increase in accuracy of 1.89% and 3.2% for the CIFAR-10 and CIFAR-100 datasets, respectively. Whereas, VGG16 demonstrated an improvement of approximately 1.8% to 2.0% in accuracy over the baseline due to EncodeNet.

4.4 Ablation Studies

Table 3. Ablation studies for the EncodeNet framework on the CIFAR-10 dataset using ResNet18 as the baseline.

Architecture Setup	Accuracy (%)
Baseline - ResNet18 without EncodeNet	91.12
ResNet 18 + no IC + AE-target is the same as input	89.26 (↓ 1.86%)
ResNet 18 + no IC + AE-lowest entropy image per class	92.11 (↑ 0.99%)
ResNet 18 + IC + AE-representative image for each sub-class	92.87 (↑ 1.75%)

To assess the impact of the different parts of our EncodeNet framework, we conducted ablation studies on the CIFAR-10 dataset using ResNet18 as the baseline DNN. Table 3 shows the said results. The baseline classification accuracy for ResNet18, without using EncodeNet, on CIFAR-10 is 91.12%. EncodeNet relies on an autoencoder for the approach to work. However, we can have different target image for feature learning. In the first setup, we set the target image for the AE as the same as the input image. By using this setup, the classification accuracy reduces to 89.26% (decrease of 1.86%), indicating ineffective feature learning for the entire dataset. In the second setup we perform entropy-based sorting of the image of a given class and use the lowest entropy image as the representative image for the entire class. EncodeNet provides improved learning of the image features, thereby, increasing the accuracy to 92.11% (increase of 0.99%). We improve the feature learning and increase the accuracy even further with the introduction of intraclass clustering. Here, instead of using one representative image for the entire class, we derive different representative images

for each of the sub-classes. Specifically, the last setup, which is the complete EncodeNet framework, leads to the classification accuracy of 92.87% (increase of 1.75%).

4.5 Comparison with Knowledge Distillation Techniques

Table 4. Comparison with various Knowledge Distillation techniques on the CIFAR-100 dataset.

Frameworks	Teacher:- Student:-	ResNet110 (74.56) ResNet 32 (71.09)	VGG16 (73.35) VGG8 (70.21)
Baseline (student)		71.09	70.21
Knowledge Distillation (KD) [14]		72.94	72.52
RKD (Relational KD) [22]		71.03	71.48
FitNet [17]		71.39	71.08
FT (Factor Transfer) [16]		73.47	71.76
EncodeNet		74.13	73.41

Knowledge Distillation approaches are known to increase the accuracy of a given baseline DNN model (Student) using a larger DNN model (Teacher). This relies on the availability of a larger Teacher model, which may not always be the case. Nonetheless, we compare our EncodeNet framework against four different knowledge techniques: KD (Knowledge Distillation) [14], RKD (Relational Knowledge Distillation) [22], FitNet [17], and FT (Factor Transfer) [16]. Table 4 shows the results for these methods for two different student-teacher combinations on the CIFAR-100 dataset. We employed two sets of teacher-student models: one with ResNet-110 as the teacher and ResNet-32 as the student, and the other with VGG16 as the teacher and VGG8 as the student. The results demonstrate that our method outperforms other knowledge distillation techniques for both ResNet and VGG networks.

4.6 Comparison with Attention Mechanism Based Techniques

Attention mechanism based techniques increase the performance of the DNN models, but that usually leads to a substantial increase in the number of model parameters. We compared EncodeNet with different attention mechanism techniques, namely Squeeze-and-Excitation Networks (SE) [23], Bottleneck Attention Module (BAM) [18], Convolutional block attention module (CBAM) [19], and Global attention mechanism (GAM) [20]. In Table 5, we observe that our framework can increase the accuracy of baseline DNN at par with other attention mechanism, but without increasing the model parameter size. For example, Global attention Mechanism (GAM) [20] shows higher improvement in classification accuracy (81.33%) compared to EncodeNet (80.10%), but significantly increases the model parameter size (approx 5 times).

Table 5. Comparison with Attention Mechanism based techniques on the CIFAR-100 dataset.

Architecture	Parameters	Accuracy (%)
ResNet 50 [1]	23.71M	77.23
ResNet 50 + SE [23]	26.22M	79.71
ResNet 50 + BAM [18]	24.06M	80.03
ResNet 50 + CBAM [19]	26.24M	80.56
ResNet 50 + GAM [20]	149.47M	81.33
ResNet 50 + GAM (gc*) [20]	57.05M	81.01
ResNet 50 + EncodeNet	23.71 M	80.10

5 Conclusion

This paper introduces a novel framework to improve the classification accuracy of baseline deep neural network (DNN) models without introducing additional parameters. Our proposed methodology outperforms recent knowledge distillation and attention mechanism approaches focused on enhancing DNN model efficacy. For each given baseline DNN model, we utilize the feature extraction layer as the encoder module of an autoencoder, design the decoder based on this feature extractor, and subsequently retrain the autoencoder to convert the given images into representative images (easy images). For robust training of autoencoder, we also implemented intraclass clustering that helped to minimize the reconstruction loss. Furthermore, we integrate the concept of knowledge transfer by employing the frozen encoder section, appending the baseline classifier, and retraining it for classification tasks. Our methodology is validated using CIFAR-10 and CIFAR-100 datasets, showcasing its capability to boost the performance of any DNN model. Additionally, we conducted a comparative analysis with existing techniques that concentrate on enhancing baseline DNN performance.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016). <https://doi.org/10.1109/CVPR.2016.90>
2. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2017)
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN (2017). <https://api.semanticscholar.org/CorpusID:54465873>
4. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149 (2015)
5. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6230–6239 (2016)

6. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 640–651 (2017)
7. Liang, T., Glossner, C.J., Wang, L., Shi, S.: Pruning and quantization for deep neural network acceleration: a survey. *Neurocomputing* **461**, 370–403 (2021)
8. Luo, J.-H., Wu, J., Lin, W.: ThiNet: a filter level pruning method for deep neural network compression. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5068–5076 (2017)
9. Aghli, N., Ribeiro, E.: Combining weight pruning and knowledge distillation for CNN compression. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2021). <https://doi.org/10.1109/CVPRW53098.2021.00356>
10. Yang, T.-J., Chen, Y.-h., Sze, V.: Designing energy-efficient convolutional neural networks using energy-aware pruning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6071–6079 (2016)
11. Zhao, Y., Li, J., Gong, Y.: Low-rank plus diagonal adaptation for deep neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2016). <https://doi.org/10.1109/ICASSP.2016.7472630>
12. Jacob, B., et al.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2704–2713 (2017)
13. Louizos, C., Ullrich, K., Welling, M.: Bayesian compression for deep learning. *arXiv preprint [arxiv: abs/1705.08665](https://arxiv.org/abs/1705.08665)* (2017)
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. *arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531)* (2015)
15. Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., Kolesnikov, A.: Knowledge distillation: a good teacher is patient and consistent. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
16. Kim, J., Park, S., Kwak, N.: Paraphrasing complex network: network compression via factor transfer. *arXiv preprint [arxiv: abs/1802.04977](https://arxiv.org/abs/1802.04977)* (2018)
17. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: FitNets: Hints for Thin Deep Nets (2015)
18. Park, J., Woo, S., Lee, J.-Y., Kweon, I.S.: Bam: Bottleneck attention module. *arXiv preprint [arXiv:1807.06514](https://arxiv.org/abs/1807.06514)* (2018)
19. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
20. Liu, Y., Shao, Z., Hoffmann, N.: Global attention mechanism: retain information to enhance channel-spatial interactions. *ArXiv preprint [arxiv: abs/2112.05561](https://arxiv.org/abs/2112.05561)* (2021)
21. Mahmud, H., Kang, P., Desai, K., Lama, P., Prasad, S.K.: A converting autoencoder toward low-latency and energy-efficient DNN inference at the edge. In: 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 592–599 (2024). <https://doi.org/10.1109/IPDPSW63119.2024.00117>
22. Park, W., Kim, D., Lu, Y., Cho, M.: Relational Knowledge Distillation (2019)
23. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
24. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., Bottou, L.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**(12) (2010)

25. Wick, C.: Deep Learning. *Informatik-Spektrum* **40**(1), 103–107 (2016). <https://doi.org/10.1007/s00287-016-1013-2>
26. Kingma, D.P., Welling, M., et al.: An introduction to variational autoencoders. *Found. Trends® Mach. Learn.* **12**(4), 307–392 (2019)
27. Mahmud, H., Kang, P., Desai, K., Lama, P., Prasad, S.: A converting autoencoder toward low-latency and energy-efficient DNN inference at the edge. *ArXiv - Accepted for publication at the PAISE workshop at IPDPS* (2024)
28. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: *IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
29. Mishra, A., Marr, D.: Apprentice: using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852* (2017)
30. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: FitNets: hints for thin deep nets. *CoRR arxiv preprint arxiv: abs/1412.6550* (2014)
31. Liu, S., Du, J., Nan, K., Zhou, Z., Wang, A., Lin, Y.: Adadeep: a usage-driven, automated deep model compression framework for enabling ubiquitous intelligent mobiles. *CoRR arxiv preprint arxiv: abs/2006.04432* (2020)
32. Lee, S., Nirjon, S.: Subflow: A dynamic induced-subgraph strategy toward real-time DNN inference and training. In: *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)* (2020). <https://doi.org/10.1109/RTAS48715.2020.00-20>
33. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017)
34. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetv2: inverted residuals and linear bottlenecks. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
35. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. *ArXiv preprint arxiv: abs/1505.04597* (2015)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *CoRR arxiv preprint arxiv: abs/1409.1556* (2014)
37. Deng, J., et al.: ImageNet: a large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2009)
38. Thorndike, R.: Who belongs in the family? *Psychometrika* **18**(4), 267–276 (1953)
39. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. In: *USENIX Symposium on Operating Systems Design and Implementation* (2016). <https://api.semanticscholar.org/CorpusID:6287870>
40. mdistiller. <https://github.com/megvii-research/mdistiller.git>
41. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. *ArXiv preprint arxiv: abs/1912.01703* (2019)
42. Krizhevsky, A., Nair, V., Hinton, G.: CIFAR-10 (Canadian institute for advanced research)
43. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)

Author Index

A

Abdullah, Ahmed 176
Adjeroh, Donald A. 160
Aggarwal, Manoj 398
Akhtar, Mushir 63
Amin, M. Ashraful 351
Arshad, Mohd. 63

B

Bahri, Omar 430
Böhm, Stefan Andreas 143
Bortiew, Amos 320
Boubrahimi, Soukaina Filali 430

C

Chakraborty, Chandan 93
Chakraborty, Debjani 304
Chatterjee, Ankita 35
Chen, Qingsheng 254
Cheung, Yiu-ming 254
Chirag, P. 1
Chirakkal, Sanid 270
Chudzik, Artur 193
Cilia, Nicole Dalia 123
Conti, Vincenzo 123

D

Das, Nabajyoti 320
Das, Partha Pratim 35
Desai, Kevin 463
Doretto, Gianfranco 160
Dutta, Sandip 35

E

Ebert, Nikolas 176

F

Fanara, Giovanni 123
Fujitomi, Taku 224

G

Garst, Swier 107
Ghosh, Mridul 93
Gong, Wenjuan 17
González, Jordi 17
Grosz, Steven 398
Guerra, Fabien 287
Guha, Debashree 304
Gupta, Pragya 304
Gupta, Ravi Kant 1

H

Halder, Nabarun 351
Hamdi, Shah Muhammad 430
Hartley, Mary-Anne 160
Hossain, Md. Shazzad 351
Hosseinzadeh, Pouya 430
Hsieh, Jun-Wei 366
Hsieh, Yi-Kuan 366

I

Islam, Ashraful 351

J

Jain, Anil 398
Jain, Sambhav 50
Jiang, Xiaoyi 208
Jin, Yi 17

K

Knoll, Alois 143
Kortüm, Christian 143

L

Lama, Palden 463
Lasker, Asifuzzaman 93
Li, Jiaxuan 414
Li, Kechen 17
Liu, Jianhao 17
Luo, Xiaopeng 254

M

Mahmud, Hasanul 463
 Makhija, Shradha 270
 Mandal, Srimanta 270
 Mandal, Subhamoy 304
 Medioni, Gerard 398
 Mishra, Sumit 335
 Moser, Gabriele 287
 Mukhopadhyay, Jayanta 35

N

Neumayer, Martin 143
 Nouyed, Mohammad Iqbal 160
 Nowacki, Jerzy P. 193

O

Obaidullah, Sk Md 93
 Oehri, Sven 176

P

Pal, Umapada 93
 Pandya, Utkarsh 270
 Pastorino, Martina 287
 Patra, Swarnajyoti 320
 Prakash, Ved 335
 Pranav, Jeevan P. 1
 Prasad, Sushil K. 463
 Przybyszewski, Andrzej W. 193
 Putrevu, Deepak 270

R

Ranjan, Rajeev 398
 Rastogi, Reshma 50
 Reinders, Marcel 107
 Reiß, Fabian 143
 Roy, Debapriya 79
 Roy, Kaushik 93

S

Sahbi, Hichem 382
 Serpico, Sebastiano B. 287
 Sethi, Amit 1
 Setu, Jahangir Hossain 351
 Shi, Tingbo 17
 Shibata, Takashi 224
 Sogi, Naoya 224
 Stricker, Didier 176
 Swain, Srinibas 335

T

Tanveer, M. 63
 Terao, Makoto 224
 Tyagi, Dinesh Kumar 446

V

Vipparthi, Santosh Kumar 446

W

Wagle, Mukta 1
 Wang, Hongcheng 398
 Wang, Jiayin 414
 Wang, Tung-Lin 366
 Wang, Xueyi 240
 Wasenmüller, Oliver 176

Y

Yadav, Prem Shanker 446

Z

Zagar, Bare Luka 143
 Zerubia, Josiane 287
 Zhang, Jiaqi 208
 Zhang, Yiqun 254
 Zhang, Yunfan 254
 Zhao, Rui 398
 Zhu, Tong 414
 Zhu, Xiaoyan 414
 Zou, Rong 254