

Asit Kumar Das · Janmenjoy Nayak ·  
Bighnaraj Naik · S. Vimal ·  
Danilo Pelusi *Editors*

# Computational Intelligence in Pattern Recognition

Proceedings of CIPR 2023

# **Lecture Notes in Networks and Systems**

**Volume 725**

## **Series Editor**

Janusz Kacprzyk , Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

## **Advisory Editors**

Fernando Gomide, Department of Computer Engineering and Automation—DCA, School of Electrical and Computer Engineering—FEEC, University of Campinas—UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering, Bogazici University, Istanbul, Türkiye

Derong Liu, Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, USA

Institute of Automation, Chinese Academy of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada

Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, Department of Electrical and Computer Engineering, KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

The series “Lecture Notes in Networks and Systems” publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

For proposals from Asia please contact Aninda Bose ([aninda.bose@springer.com](mailto:aninda.bose@springer.com)).

Asit Kumar Das · Janmenjoy Nayak ·  
Bighnaraj Naik · S. Vimal · Danilo Pelusi  
Editors

# Computational Intelligence in Pattern Recognition

Proceedings of CIPR 2023



Springer

*Editors*

Asit Kumar Das  
Department of Computer Science  
and Technology  
Indian Institute of Engineering Science  
and Technology  
Howrah, West Bengal, India

Bighnaraj Naik  
Department of Computer Applications  
Veer Surendra Sai University of Technology  
Burla, Odisha, India

Danilo Pelusi  
Department of Communication Sciences  
University of Teramo  
Teramo, Italy

Janmenjoy Nayak  
Department of Computer Science  
Maharaja Sriram Chandra Bhanja Deo  
University (MSCB)  
Baripada, Odisha, India

S. Vimal  
Department of Artificial Intelligence  
and Data Science  
Ramco Institute of Technology  
Rajapalayam, Tamil Nadu, India

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-981-99-3733-2

ISBN 978-981-99-3734-9 (eBook)

<https://doi.org/10.1007/978-981-99-3734-9>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature  
Singapore Pte Ltd. 2023

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 Singapore Pte Ltd.  
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721,  
Singapore

# **Organization Committee**

## **Chief Patrons**

Prof. (Dr.) Gautam Roy Chowdhury, Chairman, Techno India Group  
Ms. Manoshi Roy Chowdhury, Co-chairman, Techno India Group

## **Patrons**

Mr. Meghdut Roy Chowdhury, Executive Director and Chief Innovation Officer, Techno India Group  
Mr. Anit Adhikari, Director and CEO, Techno India Group  
Dr. Sujoy Biswas, Director and CEO, Techno India Group  
Dr. Avijit Kar, Director, Techno Main Salt Lake  
Dr. Kisor Roy, Director General, Techno India Group  
Mr. Soumitra Sasmal, Registrar, Techno Main Salt Lake

## **Honorary Advisory Chairs**

Prof. Lakshmi C. Jain, University of Canberra, Australia  
Prof. Michael Pecht, Chair Professor and Director, University of Maryland, College Park, USA  
Prof. (Dr.) V. E. Balas, Aurel Vlaicu University of Arad, Romania  
Prof. Ashish Ghosh, Machine Intelligent Unit, Indian Statistical Institute, Kolkata, West Bengal, India

## Honorary General Chairs

Prof. David Al-Dabass, Nottingham Trent University, UK

Prof. Jaya Sil, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India

Prof. Pabitra Mitra, Indian Institute of Technology, Kharagpur, West Bengal, India

Prof. Paramartha Dutta, Visva-Bharati, Santiniketan, West Bengal, India

## General Chairs

Prof. Asit Kumar Das, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India

Prof. Saroj K. Meher, Indian Statistical Institute, Bangalore Centre

Prof. Weiping Ding, Nantong University, Jiangsu, China

Prof. Tanmay De, National Institute of Technology, Durgapur, West Bengal, India

## Program Chairs

Dr. Janmenjoy Nayak, Maharaja Sriram Chandra Bhanja Deo (MSCB) University, Baripada, Mayurbhanj, Odisha, India

Dr. Danilo Pelusi, University of Teramo, Coste Sant'agostino Campus, Teramo, Italy

Dr. S. Vimal, Ramco Institute of Technology, Tamil Nadu, India

Dr. Bighnaraj Naik, Veer Surendra Sai University of Technology, Burla, Odisha, India

## Co-program Chairs

Dr. Malay Kule, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India

Dr. Nirnay Ghosh, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India

Dr. Santanu Phadikar, Maulana Abul Kalam Azad University of Technology, West Bengal, India

Dr. Soumi Dutta, IEM, West Bengal, India

## Organizing Chairs

Ms. Poulami Dutta, Department of CSE, Techno Main Salt Lake  
Dr. Tapan Chowdhury, Department of CSE, Techno Main Salt Lake  
Ms. Nairanjana Chowdhury, Department of CSE, Techno Main Salt Lake  
Dr. Kaustubh Dutta, Department of Mathematics, Techno Main Salt Lake  
Dr. Sovik Roy, Department of Mathematics, Techno Main Salt Lake  
Mr. Mrinal Kanti Nath, Department of CSE, Techno Main Salt Lake  
Ms. Manashi De, Department of CSE, Techno Main Salt Lake

## International Advisory Committee

Dr. Florin Popentiu Vladicescu, University of Oradea, Romania  
Dr. Arijit Sur, IIT Guwahati, India  
Dr. Charlie (Seungmin) Rho, Chung-Ang University, Seoul, Korea  
Prof. Shaikh A. Fattah, Bangladesh University of Engineering and Technology, Bangladesh  
Dr., Claude Delpha, Université Paris Saclay, France  
Prof. Pabitra Mitra, IIT Kharagpur, India  
Dr. Jong Hyuk Park, Seoul National University of Science and Technology, Korea  
Dr. Mohammad S. Khan, East Tennessee State University, USA  
Prof. Susanta Chakraborty, IEST, Shibpur, Howrah, West Bengal, India  
Dr. Rubén González Crespo, Universidad, Internacional de La Rioja  
Dr. Sheng-Lung Peng, National Dong Hwa University, Hualian, Taiwan  
Dr. Swagatam Das, Indian Statistical Institute, Kolkata, India  
Prof. Raffaele Mascella, University of Teramo, Italy  
Prof. Sulata Mitra, IEST Shibpur, West Bengal, India  
Dr. Prabhat Kumar, National Institute of Technology Patna, India  
Prof. Debdatta Sinha, University of Calcutta, India  
Prof. K. C. Santosh, University of South Dakota  
Prof. Luca Tallini, University of Teramo, Italy  
Dr. Atanu Bhattacharjee, Homi Bhabha National Institute Section of Biostatistics, India  
Dr. Shahid Mumtaz, Instituto de Telecomunicaciones, Aveiro, Portugal  
Prof. Monojit Mitra, IEST, Shibpur, West Bengal  
Prof. Yong Deng, Institute of Fundamental and Frontier Science Chengdu, China  
Prof. Rashmi Gupta, Netaji Subhas University of Technology, East Campus, India  
Prof. Amir H. Gandomi, University of Technology Sydney, Australia  
Prof. Mita Nasipuri, Jadavpur University, West Bengal, India  
Dr. Xiao-Zhi Gao, University of Eastern Finland, Kuopio, Finland  
Prof. Santi Prasad Maity, IEST, Shibpur, West Bengal, India  
Prof. Paramartha Dutta, Visva Bharati University, West Bengal, India

Prof. Alireza Souri, Islamic, Azad University: Sardroud, IR  
Dr. Robert Bestak, Czech Technical University in Prague, Czech Republic  
Prof. Qin Xin, University of the Faroe Islands, Denmark  
Dr. Govindarajan Kannan, Indiana University Bloomington, Bloomington  
Prof. Manju Khari, JNU, New Delhi, India  
Dr. Naveen Chilamkurti, La Trobe University, Melbourne, Australia  
Prof. Gajendra K. Vishwakarma, IIT (ISM) Dhanbad, India  
Prof. Joy Iong-Zong Chen, Da-Yeh University, Taiwan  
Prof. Amitava Chatterjee, Jadavpur University, West Bengal  
Prof. Subramaniam Ganesan, Oakland University, USA  
Prof. D. P. Mohapatra, NIT Rourkela, India  
Dr. Ahmed A. Elngar, Beni-Suef University, Egypt  
Prof. Damien Sauveron, Université de Limoges, France  
Prof. Ali Kashif Bashir, Manchester Metropolitan University, UK  
Prof. B. Annappa, NIT Surathkal, Karnataka  
Prof. Victor Hugo C. de Albuquerque, University of Fortaleza, Brazil  
Prof. Chandan Kumar Chanda, IIEST, Shibpur, West Bengal  
Prof. Dac-Nhuong Le, Haiphong University, Haiphong, Vietnam  
Prof. Mamoun Alazab, Charles Darwin University, Australia  
Dr. H. S. Behera, VSSUT, Burla, Odisha, India  
Dr. Daniel Burgos, International University of La Rioja (UNIR), Spain  
Dr. Seifedine Kadry, Beirut Arab University, Lebanon  
Dr. Y. Harold Robinson, VIT University, India  
Prof. J. K. Mandal, University of Kalyani, West Bengal, India  
Dr. Xuan Liu, Future Network Research Center, Southeast University, China  
Dr. M. Kaliappan, Ramco Institute of Technology, India  
Dr. Swapnoneel Roy, University of North Florida, USA

## Technical Committee

Dr. Joy Iong-Zong Chen, Da-Yeh University, Taiwan  
Dr. S. K. Hafizul Islam, IIIT, Kalyani  
Dr. Uttam Ghosh, Meharry School of Applied Computer Sciences, Meharry Medical College, USA  
Dr. Ananya Barui, Center of Healthcare Science and Technology, IIEST, Shibpur, India  
Dr. Ahmed Elngar, Faculty of Computers and Artificial Intelligence, Beni-Suef University, Egypt  
Dr. P. Subbulakshmi, VIT University, Chennai, India  
Dr. Tanmay De, National Institute of Technology, Durgapur, West Bengal, India  
Dr. Dac-Nhuong Le, Haiphong University, Haiphong, Vietnam  
Dr. G. T. Chandra Sekhar, Sri Sivani College of Engineering, Srikakulam, Andhra Pradesh, India

- Dr. Noor Zaman, Taylor's University, Malaysia  
Dr. Rajendrani Mukherjee, University of Engineering and Management, Kolkata, West Bengal  
Dr. Irfan Mehmood, University of Bradford, UK  
Dr. L. Ganesan, Ramco Institute of Technology, India  
Dr. Gaurav Dhiman, Government Bikram College of Commerce, Patiala, India  
Dr. Arif Sari, Girne American University, UK  
Dr. Ram Sarkar, Jadavpur University, West Bengal, India  
Dr. Pradeepa, Sastra University, India  
Dr. Xiao-Zhi Gao, University of Eastern Finland, Kuopio, Finland  
Dr. Soumya Ranjan Nayak, Amity University, Noida, Uttar Pradesh, India  
Dr. Khan Muhammad, Sejong University, Seoul  
Dr. Vijay Bhaskar Semwal, MANIT Bhopal, India  
Dr. Hoang Viet Long, People's Police University of Technology and Logistics, Bac Ninh, Vietnam  
Dr. Suparna, Biswas (Saha), MAKAUT, West Bengal  
Dr. Surajeet Ghosh, IEST, Shibpur, Howrah, India  
Dr. Carla M. A. Pinto, Adjunct Professor at ISEP—Instituto Superior de Engenharia do Porto, Portugal  
Dr. Nibaran Das, Jadavpur University, West Bengal, India  
Dr. J. C. Bansal, South Asian University, New Delhi, India  
Dr. Ramani Kannan, Universiti Teknologi PETRONAS  
Dr. Samit Biswas, IEST, Shibpur, Howrah, India  
Dr. Nevine Makram Labib, Sadat Academy for Management Sciences, Egypt  
Dr. J. V. Anchitaalagammai, Velammal College of Engineering and Technology, India  
Dr. Imon Mukherjee, Indian Institute of Information Technology, Kalyani, West Bengal  
Dr. Sarat Chandra Nayak, CMR College of Engineering and Technology, Hyderabad, India  
Dr. A. Suresh, SRM University Chennai, India  
Dr. A. R. Routray, F. M. University, Odisha, India  
Prof. Alex Khang, Information Technology, Leading Expert of Data Engineering, SEFIX, Vietnam  
Dr. Chitrangada Das Mukhopadhyay, Center of Healthcare Science and Technology, IEST, Shibpur  
Dr. U. D. Prasan, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh  
Dr. Sudhakar Ilango, VIT University, Andhra Pradesh, India  
Dr. Oishila Bandyopadhyay, Indian Institute of Information Technology, Kalyani, West Bengal  
Dr. Thinagaran Perumal, Universiti Putra Malaysia, Malaysia  
Dr. V. Jackins, National Engineering College, India  
Dr. B. Acharya, National Institute of Technology Raipur, India  
Dr. Golden Julie, Anna University Tirunelveli, India

Dr. Jeyabalara, Velammal Engineering College, India  
H. Swapnarekha, Aditya Institute of Technology and Management, Andhra Pradesh, India  
Dr. Dac-Nhuong Le, Faculty of Information Technology, Haiphong University, Vietnam  
Dr. L. Jerart Julus, National Engineering College, India  
Dr. Ronnie Figueiredo, Universidade da Beira Interior—UBI, Portugal  
Dr. Ranjit Ghosal, St. Thomas College of Engineering and Technology, Kolkata, West Bengal

## **Web Chair**

Dr. P. Suresh Kumar, Department of Computer Science and Engineering, GITAM (Deemed to be University), Visakhapatnam, India

## **Publicity Chairs**

Dr. Rahul Das Gupta, Department of CSE, Techno Main Salt Lake  
Dr. Abhishek Das, Department of CSE, Aliah University, Kolkata  
Mr. Utpal Das, Department of CSE, Techno Main Salt Lake  
Mr. Avijit Mondal, Department of CSE, Techno Main Salt Lake  
Dr. Naela Rizvi, Department of CSE, Techno Main Salt Lake  
Ms. Devleena Ghosh, Department of CSE, Techno Main Salt Lake

## **Publication Chairs**

Dr. Sunanda Das, Department of Computer Science and Engineering, Jain University, Bangalore, India  
Dr. Arpita Biswas, Department of CSE, Techno Main Salt Lake  
Dr. Shampa Sengupta, Department of IT, MCKVE, Howrah, West Bengal  
Dr. Ranjit Ghoshal, Department of CSE, St. Thomas College of Engineering and Technology, Kolkata, India

## **Finance Chairs**

Prof. Asit Kumar Das, IIEST, Shibpur, Howrah, West Bengal, India  
Dr. Malay Kule, IIEST, Shibpur, Howrah, West Bengal, India

## CIPR Reviewers

- Dr. K. V. Uma, Thiagarajar College of Engineering, Madurai  
Dr. P. V. Siva Kumar, VNR VJIET, Hyderabad  
Sourav Das, Future Institute of Technology, Kolkata  
Poly Ghosh, Primeasia University, Banani, Dhaka, Bangladesh  
Dilip Kumar Dalei, Defence Research and Development Organisation (DRDO), Bengaluru  
Dr. Bhaveshkumar C. Dharmani, Lovely Professional University (LPU), Punjab  
Dr. P. M. K. Prasad, GVP College of Engineering for Women, Visakhapatnam  
Hiral M. Patel, Sankalchand Patel College of Engineering, Gujarat  
Dr. Ripal D. Ranpara, Atmiya University, Gujarat  
Dr. Mainak Bandyopadhyay, KIIT Deemed to be University, Bhubaneswar  
Dr. S. Rama Sree, Aditya Engineering College, Surampalem  
Karun Kumar Reddy, Dr. Lankapalli Bullayya College of Engineering, Visakhapatnam, Andhra Pradesh  
Dr. Soumya Ranjan Nayak, Amity University, Noida  
Mr. Byomakesha Das, Aditya Institute of Technology and Management, Tekkali  
Dr. Janmenjoy Nayak, Maharaja Sriram Chandra Bhanja Deo (MSCB) University, Baripada, Mayurbhanj, Odisha  
Dr. Manohar Mishra, Department of Electrical and Electronics Engineering, Siksha O Anusandhan (Deemed to be University), Bhubaneswar  
Dr. Bighnaraj Naik, Veer Surendra Sai University of Technology, Burla  
Dr. Malay Kule, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal  
Dr. Bhaskar Patnaik, Malla Reddy engineering College, Hyderabad  
Dr. Rasmi Ranjan Panigrahi, Gandhi Engineering College, Bhubaneswar  
Dr. S. Vimal, Ramco Institute of Technology, Tamil Nadu  
Dr. Ram Barik, Vikash Institute of Technology, Bargarh  
Dr. P. Suresh Kumar, GITAM University, Visakhapatnam  
H. Swapnarekha, Aditya Institute of Technology and Management, Tekkali  
Dr. Asit Kumar Das, IIEST, Shibpur  
Mrs. Subhasree Mohapatra, ITER, SOA University, Bhubaneswar, Odisha  
Dr. Sharmila Subudhi, Maharaja Sriram Chandra Bhanja Deo University, Baripada, Odisha  
Ranit Kumar Dey, IIEST, Shibpur  
Dr. Tapas Kumar Mishra, SRM University, Andhra Pradesh  
Amrit Chhetri, Digital Forensic Researcher and CEI (Rosefinch, Siliguri)  
Dr. Dinesh Kumar Nali, Vignan Institute of Technology and Science, Deshmukhi, Hyderabad  
Dr. Ankur Kumar Shrivastava, Senior Specialist Information Security RTA Dubai, UAE,  
Boudhayan Bhattacharya, Brainware University, West Bengal  
Prof. Umesh C. Pati, National Institute of Technology, Rourkela, Odisha, India

- Dr. G. L. N. Murthy, LBR College of Engineering, Andhra Pradesh  
Dr. R. Ramya, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu  
Giridhar Maji, Asansol Polytechnic, Department of Technical Education and Training, West Bengal  
Dr. Ram Ratan, Scientist 'G' (retd.), Member IDST, DRDO, Delhi, India  
Swapnil Singh, Mukesh Patel School of Technology Management and Engineering, NMIMS University, Mumbai  
Gireesha O., CBIT, Hyderabad  
Karthickmanoj R., AMET Deemed to be university, Chennai  
Antonio Marceddu, Dipartimento di Automatica e Informatica, Politecnico di Torino, C.so Duca degli Abruzzi 24 10129 Torino, Italy  
Dr. M. Sundar Prakash Balaji, Mookambigai College of Engineering, Pudukkottai  
Dr. Nirav H. Bhatt, Charotar University of Science and Technology, Changa  
Dr. Nikita Bhatt, Charotar University of Science and Technology, Changa  
Dr. Sushma Jaiswal, Computer Science and Information Technology (CSIT), Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur (C.G.)  
Dr. Apurva Mehta, Dharmsinh Desai University-Nadiad, Gujarat  
Dr. Rohini Hallikar, R. V. College of Engineering, Bengaluru, Karnataka  
Pavate Aruna Anilish, St. Francis Institute of Technology, Mumbai Maharashtra India.  
Dr. Anand Kumar, Jagadguru Rambhadracharya Divyanga University, Chitrakoot, Uttar Pradesh  
Dr. Arati J. Vyawahare, PES'S Modern college of Engineering, Pune  
Salil Bharany, School of Computer Engineering and Technology, Punjab, India  
Dr. P. Sirish Kumar, Aditya Institute of Technology and Management, Tekkali  
Ila Pavan Kumar, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad  
Dr. Sunil, Jamia Millia Islamia (A Central University), New Delhi  
Dr. S. Hari Haran, Vardhaman College of Engineering, Hyderabad, Telangana  
Dr. Kanika Lakhani, The M. S. University of Baroda, Vadodara  
Ghanta Sai Krishna, IIIT, Naya Raipur  
Rajeev Ratna Vallabhuni, Bayview Asset Management, LLC  
Dr. T. Jemima Jebaseeli, Karunya Institute of Technology and Sciences, Coimbatore  
Surya Prakash Y., GMR Institute of Technology, Rajam

# Preface

Computational intelligence (CI) is a rapidly evolving field that encompasses a range of cutting-edge technologies, such as fuzzy logic, higher-order neural networks, swarm and memetic computing, deep learning, and hybrid models, among others. These technologies are essential for developing intelligent systems, including games and mental developmental systems, which have become increasingly important in recent years. The growing volume of data has led to a significant demand for intelligent computing in various scientific and technological fields. This demand is expected to contribute to the overall advancement of science and technology and improve the quality of life. Researchers have increasingly adopted CI techniques, such as deep learning (DL), neural networks (NN), fuzzy logic (FL), evolutionary algorithms (GA), and other CI methods to develop efficient models that incorporate various intelligent techniques to solve complex pattern recognition problems.

We are pleased to extend a warm welcome to the 5th International Conference on Computational Intelligence in Pattern Recognition (CIPR), scheduled to be held on the 27th and 28th of May 2023 at Techno Main Salt Lake in Kolkata, West Bengal, India. Our primary objective is to bring together academic researchers, engineers, and industry professionals to share and exchange their experiences and research results on various aspects of science and social exploration. The conference aims to provide a platform for experts to present and discuss the latest advancements, trends, concerns, and practical challenges encountered in the fields of pattern recognition and intelligent computing. This proceeding is intended to provide rapid dissemination of significant results and high-level concepts in the latest domains of intelligent computing, deep learning, soft computing, etc. We are honored to present the proceedings of CIPR 2023 to the authors and representatives of the event. We are confident that the conference will be beneficial, motivating, and inspiring for all the participants. CIPR 2023 promises to be an enriching experience with a fantastic line-up of keynote speakers from around the world. The conference will provide a platform to explore the issues, challenges, opportunities, and discoveries of computational intelligence and recognition research. The constantly evolving scope and rapid advancement of intelligent techniques create new issues and uncertainties, underscoring the need for sharing brilliant, inspiring ideas, and stimulating thoughtful discussions in this vast

research field. We pledge to create an engaging atmosphere for pattern recognition, and the overwhelming support and excitement we have received have exceeded our expectations.

The 5th edition of the CIPR conference welcomed proposals on the implementation of computational intelligence in diverse areas such as text and video identification, opinion analysis, and advanced image processing. The conference will feature articles that cover major and minor thematic areas, with a wide range of topics including prediction, stock exchange analysis, real-time video analysis, text recognition, language recognition, fingerprint analysis, patient analysis and monitoring, concrete crack recognition, cancer analysis and detection, and student lifestyle query categorization. Each paper submitted to the conference is reviewed by subject specialists and the editorial team, in collaboration with international advisory, program, and technical committee members, to ensure its quality and relevance.

CIPR 2023 is a conference that includes 59 high-quality papers, which were chosen through a double blind peer-review process involving the technical committee and international members. The conference provided a platform for sharing knowledge and expertise among research communities from different countries. The accepted papers have been classified according to their focus on the latest trends in computational intelligent techniques in pattern recognition. The authors' contributions have added significant value to the conference. The success of CIPR 2023 is due to the collective efforts of the authors, organizing committee, and publications. The conference presents informative contributions for scholars worldwide, covering novel and innovative methods, state-of-the-art techniques, and applications in research fields. We extend our gratitude to the keynote and panel speakers, committee members, advisory board members, technical committee members, and reviewers for their invaluable support. We also acknowledge the technical team from Springer for their collaboration in publishing the conference proceedings. We hope that you find the conference fruitful and enjoyable, and your contributions are essential to its success.

Howrah, India

Baripada, India

Burla, India

Rajapalayam, India

Teramo, Italy

Asit Kumar Das

Janmenjoy Nayak

Bighnaraj Naik

S. Vimal

Danilo Pelusi

# Acknowledgements

With great pleasure, we present this volume of carefully curated papers on computational intelligence in pattern recognition (CIPR). The fifth version of CIPR had faith in high-quality areas of computational insight-based research and developments after the successful four earlier versions. This version attracted a few academicians or researchers from around the world to select this venue for presenting the papers and give the CIPR 2023 meeting a broad height for research findings and sharing the information between the general public and international specialists. The program consists of warmly invited gatherings, technical workshops, and discussions with eminent speakers covering a wide scope of subjects in science and social exploration. Each participant has the opportunity to get to know and interact with others through this rich program. We are confident that your participation in CIPR 2023 will be fruitful and enduring.

More than 200 academicians, scholars, or analysts from all over the world have expressed interest in the CIPR meeting's vital proposal and significance. They have been urged to submit papers of higher quality and to demonstrate the meeting's standing for original examination disclosures, the pattern of consideration, and the imparting of information to both national and international cooperatives in various fields and angles of information investigation, and pattern recognition goes out to everyone who presented their planned research at CIPR. The general organizing committee would like to express its gratitude for the support and solace received from our committee and the many others who helped plan this event.

We wanted to express our appreciation to our authors, whose important research findings made this event great and welcomed speakers, presenters, and audiences as well. We want to express our sincere gratitude to the CIPR organizing team from Techno Main Salt Lake in Kolkata for their invaluable assistance in setting up all of the systems necessary to make this international event a success. We would like to heartfully thank the beloved Director and Registrar of Techno Main Salt Lake in Kolkata for their constant guidance and support throughout the planning of the CIPR 2023 conference. In addition to these individuals, we also like to express our gratitude to the departmental staff members who made a significant contribution to the success of the conference.

We have had the good fortune to collaborate with outstanding members of national and international advisory, technical, and program committees. The participants and the members of the expert panel were enthusiastic about the quality outcomes from the very first day of the event, and their suggestions have created all feasible means of distributing high-quality articles throughout all submitted papers. We may like to extend our sincere gratitude to the reviewers. They have worked diligently to review papers and develop critical suggestions for authors to enhance their work. We would like to express our deep gratitude and genuine appreciation to the editorial members of Springer Publishing for not only working with us to complete the proceedings but also for their timely and impeccable outfit distributions that follow a sophisticated, development-focused strategy for the successful intime publication.

Asit Kumar Das  
Janmenjoy Nayak  
Bighnaraj Naik  
S. Vimal  
Danilo Pelusi

# Contents

<b>A New Technique of Cipher Type Identification Using Convolutional Neural Networks .....</b>	1
Subinoy Sikdar and Malay Kule	
<b>Monthly Rainfall Forecasting Using Sequential Models .....</b>	17
A. Kala, P. Sharon Femi, V. Rajalakshmi, and K. Ashwini	
<b>Detection and Classification of Dental Caries Using Deep and Transfer Learning .....</b>	27
Divya Rajput, Hiral Rane, Devika Nikam, Janhavi Wagh, and Anuja Jadhav	
<b>Identification of Diabetic Retinopathy Using Robust Segmentation Through Mask RCNN .....</b>	39
Aryan and Suman Deb	
<b>Digital Watermarking Using Visual Cryptography .....</b>	49
Alina Dash, Kshiramani Naik, and Sharmila Subudhi	
<b>Analytical Comparison of Deep Learning Frameworks for Semantic Segmentation with Pixel-Level Understanding .....</b>	63
Ankit Deb, Shuvrajeet Das, and Suman Deb	
<b>Method-Level Code Smells Detection Using Machine Learning Models .....</b>	77
Seema Dewangan and Rajwant Singh Rao	
<b>Q-Learning-Based Node Scheduling for Energy Saving in WSN .....</b>	87
Jyoti and Tamal Pal	
<b>Performance Investigation of SVM and Modified SVM Algorithms for Acute Health Diagnosis .....</b>	97
Pallavi Sharma, Rajni Bedi, and Vikram Dhiman	

<b>An Efficient Multifactor Authentication System .....</b>	109
Shreya Verma, Mansi Singh, Krittika Chaturvedi, and B. K. Tripathy	
<b>A Decision-Based Image Merging Technique for Server-Side Redundancy Reduction .....</b>	123
Richa Kumari Kora and Tamal Pal	
<b>Handling Class Imbalance Problem Using Support Vector Machine .....</b>	133
Mehwish Naushin, Ankur Das, and Asit Kumar Das	
<b>Gender and Hand Identification Based on Dactyloscopy Using Deep Convolutional Neural Network .....</b>	145
Diptadip Maiti and Debashis Das	
<b>A CNN-Based Approach for Face Recognition Under Different Orientations .....</b>	157
R. Ahila Priyadarshini, S. Hariharan, and R. Jagadeeswara	
<b>A Deep Learning Approach for Detection of Disease in Plant Leaves .....</b>	169
S. Khatri, B. K. Tripathy, K. V. S. S. Kumar, V. Kumar, S. Bharti, A. Gupta, and Y. Khandelwal	
<b>Label Consistency-based Modified Sequential Dictionary Learning-based Approach for PIR Sensor-based Detection of Human Movement Direction .....</b>	183
Pubali De, Amitava Chatterjee, and Anjan Rakshit	
<b>Analyzing Lung Diseases Using CNN from Chest X-ray Images .....</b>	197
Sanhita Dan, Arpan Garai, and Samit Biswas	
<b>Stance Classification on FIFA World Cup Using Twitter Data .....</b>	209
Aaquib Asrar, Susmita Das, and Sangita Dutta	
<b>Machine Learning-Based Phishing Detection in Heterogeneous Information Network .....</b>	221
Priti Halder, Ankan Mallick, Shantonu Debnath, and Malay Kule	
<b>Unified Feature Extraction for Handwritten and Natural Image Characters: A Case Study Using Bank Legal Check Amount and Vehicle License Plate Number Recognition .....</b>	235
M. Arun, S. Arivazhagan, P. Sivaramapandian, and S. Sivavakisan	
<b><math>\alpha</math> and <math>\beta</math>-Testing of an Epileptic Seizure Detection Algorithm on Pre-ictal, Ictal, and Inter-ictal Part of EEG Signal .....</b>	247
Khakon Das and Ashish Khare	
<b>SBGAN: Sequential Bengali Word Image Generation Model .....</b>	261
Piyush Kanti Samanta, Arpita Dutta, and Samit Biswas	

<b>A Transfer Learning Approach to Indian Currency Coin Recognition .....</b>	273
R. Ahila Priyadharshini, M. Vijayraj, and R. Dhanushraj	
<b>SDinIWTrack: A Novel Database for Training Self-driving Vehicles .....</b>	283
Swati Chowdhuri, Sriparna Banerjee, and Supriya Mondal	
<b>Detection of Cipher Types Using Machine Learning Techniques .....</b>	297
Abhiroop Mukherjee, Arnab Sen, Krishnendu Bera, Rajdeep Ghosh, Swarnali Mondal, Sanjana Chakravarty, Subinoy Sikdar, and Malay Kule	
<b>An Overview of Segmentation Models for the Extraction of Brain Tissues from Magnetic Resonance Images .....</b>	309
Elisabeth Thomas and S. N. Kumar	
<b>Detection of Cyberattacks in Cyber-Physical Systems Using Supervised Learning and Hypergraphs .....</b>	323
S. Priyanga, S. Pravinraj, Venkata Bhavana Repalle, Kannan Krishivasan, and V. S. Shankar Sriram	
<b>Classifying Fetal Health Using Neural Networks by Boosting Imbalanced Classes .....</b>	337
Perumalla Anoosha, Renuka Devi Parlapalli, E. Srikanth Reddy, and P. Menaga	
<b>Cryptonate: Crypto-Jacking Web Extension .....</b>	347
Nilesh Patil, Dhruv Gandhi, Pranay Prajapati, and Kevin Haria	
<b>Real-Time Mood-Based Music Auto-Play System from Facial Expressions .....</b>	363
Raunaq Singh, Vipin Singh, Pooja Verma, G.V. Eswara Rao, and Rajitha Bakthula	
<b>A Wrapper-based Feature Selection Approach Using Particle Swarm Optimization for Software Fault Prediction .....</b>	375
Hritik Shah and Himansu Das	
<b>Sign Language to Sentence Interpreter Using Convolutional Neural Network in Real Time .....</b>	387
S. Seetha, C. Christlin Shanuja, Esther Daniel, Saurabh Chandra, and Saurabh Raj	
<b>A Comparative Study of Deep Learning-Based Face Recognition and Emotion Detection Techniques Using Social Media Customized Cartoon Post .....</b>	401
Jayanta Paul, Anuska Roy, Siddhartha Mallick, and Jaya Sil	
<b>A Hybrid GA-PSO based approach for Mining Top-Ranked Web Pages to Reorganize Websites .....</b>	413
Santosh Kumar, Tejas Kesarwani, and Sumit Kumar	

<b>Addressing Class Imbalance in Fake News Detection with Latent Space Resampling .....</b>	427
Saranya Bhattacharjee, Soumyajit Maity, and Sankhadeep Chatterjee	
<b>MEDNet-Based Imbalanced Cataract Detection Using Ophthalmic Images .....</b>	439
Soumyajit Maity, Saranya Bhattacharjee, Ankur Das, and Sankhadeep Chatterjee	
<b>A Supervised Learning Algorithm for Disease Prediction Using Complex Network and Machine Learning .....</b>	453
Sangita Dutta, Navnee Singh, Susmita Das, and Susanta Chakraborty	
<b>Deep Learning-Based Real-Time Hand Gesture Recognition Using Histogram of Oriented Gradient .....</b>	465
Anurag Sahu, Tannistha Pal, and Suman Deb	
<b>A Combined Approach of Color Correction and Homomorphic Filtering for Enhancing Underwater Images .....</b>	475
R. Ahila Priyadarshini and K. Ramajeyam	
<b>An Effective Pipeline for Depth Image-Based Hand Gesture Recognition .....</b>	489
Taniya Sahana and Ayatullah Faruk Mollah	
<b>A Coupled System for Simultaneous Image Despeckling and Segmentation .....</b>	505
Ankit Kumar and Subit K. Jain	
<b>An Improved Intrusion Detection System for the Internet of Medical Things Based on Deep Convolutional Neural Network .....</b>	517
Pandit Byomakesha Dash, H. S. Behera, and Manas Ranjan Senapati	
<b>Deep Convolutional Neural Network for Skin Cancer Classification .....</b>	529
Pandit Byomakesha Dash, Ch Ravi Kishore, Venkatasai Kommu, Vysyaraju Lokesh Raju, and Subhasree Mohapatra	
<b>Prediction of Absenteeism at the Workplace: A Light Gradient Boosting Approach .....</b>	543
Suresh Kumar Pemmada and Janmenjoy Nayak	
<b>Deep Neural Networks Scheme-Based Intelligent IDS System for Securing Internet of Vehicles .....</b>	555
Ch. Ravi Kishore, D. Chandrasekhar Rao, and H. S. Behera	
<b>Low-Memory Pedestrian Detection Using Binarized Neural Networks .....</b>	567
Mainak Bandyopadhyay and Rakesh Baral	

<b>A Mind-Driven Artificial Limb Movement Framework Using Long Short-Term Memory Algorithm .....</b>	583
Ahona Ghosh and Sriparna Saha	
<b>Integration of Blockchain Technology with Renewable Energy for Sustainable Development: Issues, Challenges and Future Direction .....</b>	595
Saumendra Das, H. Swapnarekha, and S. Vimal	
<b>Binary Classification of Kidney Glomeruli Using Deep Neural Networks .....</b>	609
Basra Jehangir, Soumya Ranjan Nayak, and Shaiq Wani	
<b>Application of Modified Differential Evolution Technique for Automatic Generation Control Problem .....</b>	623
Asish Kumar Panigrahi, Rabindra Kumar Sahu, and Tulasichandra Sekhar Gorripotu	
<b>A Four-Valued Epistemic Logic for Metadata Modelling from Medical Articles on Pain Therapies .....</b>	631
Simone Cuconato	
<b>Enhancing IoT Network Security with Light Gradient Boosting Machine and Gravitational Interaction Optimization for Malicious Access Detection .....</b>	641
Geetanjali Bhoi, Bighnaraj Naik, and Etuari Oram	
<b>Anti-lock Braking System Using Monte Carlo Simulations .....</b>	653
Ibidun Christiana Obagbuwa, Vincent Mohale Zibi, and Mishi Makade	
<b>Artificial Intelligence Applied to the Geography: A Connectionist Approach .....</b>	673
Mauro Preda	
<b>Artificial Intelligence, Administrative Proceeding, Protection and Enhancement of Cultural Property .....</b>	685
Livio Perra	
<b>Personalized Surgical Planning in Liver Surgery Using Virtual 3D-Models .....</b>	695
Teresa Perra and Alberto Porcu	
<b>Impact of AI on Student's Research and Writing Projects .....</b>	705
Joan Rosselló-Geli	
<b>Spaces, Videogames and Artificial Intelligence: A Geographical Approach .....</b>	715
Gaetano Sabato and Francesco De Pascale	

- S-Transform and Bayesian-Optimized Decision Tree-Based  
Islanding Detection Approach for Distributed Generation System . . . . . 727**  
Pratyush Kumar Muni, Manohar Mishra, Chinmoy Kumar Patra,  
Debadatta Amaresh Gadanayak, and Tanmoy Parida

# Editors and Contributors

## About the Editors

**Asit Kumar Das** is working as Professor in the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India. He has published more than 150 research papers in various international journals and conferences, 3 books, and 6 chapters. He has worked as Member of the Editorial/Reviewer Board of various international journals and conferences. He has shared his research field of interest in many workshops and conferences through his invited speech in various institutes in India. He acts as General Chair, Program Chair, and Advisory Member of committees of many international conferences. He has more than twenty years of teaching experience in the field of Computer Science and Technology. His research interest includes machine learning and pattern recognition in various fields including bioinformatics, social networks, text mining, audio and video data analysis, and medical data analysis. He has already guided ten Ph.D. scholars and is currently guiding six Ph.D. scholars.

**Janmenjoy Nayak** is working as Assistant Professor, P.G. Department of Computer Science, Maharaja Sriram Chandra Bhanja Deo (MSCB) University, Baripada, Odisha, India. He has published more than 210+ research papers in various reputed peer-reviewed referred journals, international conferences, and chapters. Being two times Gold Medalist in Computer Science in his career, he has been awarded with INSPIRE Research Fellowship from Department of Science and Technology, Government of India (both as JRF and SRF level) and Best Researcher Award from Jawaharlal Nehru University of Technology, Kakinada, Andhra Pradesh, for the AY: 2018–19, and many more awards to his credit. He has edited 20+ books and 15+ special issues in various topics including data science, machine learning, and soft computing with reputed international publishers like Springer, Elsevier, Inderscience, etc. His area of interest includes data mining, nature-inspired algorithms, and soft computing.

**Bighnaraj Naik** is Assistant Professor in the Department of Computer Applications, Veer Surendra Sai University of Technology, Burla, Odisha, India. He received his Doctoral degree from the Department of Computer Science Engineering and Information Technology, Veer Surendra Sai University of Technology, Burla, Odisha, India, Master's degree from SOA University, Bhubaneswar, Odisha, India, and Bachelor's degree from National Institute of Science and Technology, Berhampur, Odisha, India. He has published more than 190+ research papers in various reputed peer-reviewed international conferences, referred journals, and chapters. He has more than ten years of teaching experience in the field of Computer Science and Information Technology. His area of interest includes data mining, soft computing, etc. Currently, he is guiding four Ph.D. scholars and six master students.

**S. Vimal** is working as Associate Professor in the Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Tamil Nadu, India. He received Ph.D. degree in Cognitive Radio Networking and Security Techniques using AI from Anna University, Chennai, Tamil Nadu. He is working as Associate Professor in the Department of Computer Science and Engineering, Ramco Institute of Technology, Tamil Nadu, India. His areas of interest include game modeling, artificial intelligence, cognitive radio networks, and network security. He has published around 80 papers. He has hosted 21 special issues in IEEE, Elsevier, Springer, and CMC Tech Science journals.

**Danilo Pelusi** received the degree in Physics from the University of Bologna (Italy) and the Ph.D. degree in Computational Astrophysics from the University of Teramo (Italy). Currently, he is Associate Professor of Computer Science at the Department of Communication Sciences, University of Teramo. Editor of Springer and Elsevier books, and Associate Editor of *IEEE Transactions on Emerging Topics in Computational Intelligence* (2017–2020), *IEEE Access* (2018–present) and *IEEE Transactions on Neural Networks and Learning Systems* (2022–present), *IEEE Transactions on Intelligent Transportation Systems* (2022–present), he is Guest Editor for Elsevier, Springer, MDPI, and Hindawi journals. Keynote Speaker, Guest of Honor, and Chair of IEEE conferences, he is Inventor of international patents on artificial intelligence. His research interests include fuzzy logic, neural networks, information theory, machine learning, and evolutionary algorithms.

## Contributors

**Perumalla Anoosha** School of Computer Science and Artificial Intelligence, SR University, Warangal, Telangana, India

**S. Arivazhagan** Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

**M. Arun** Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

**Aryan** Department of CSE, NIT Agartala, Paschim Barjalai, India

**K. Ashwini** Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India

**Aaquib Asrar** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India

**Rajitha Bakthula** Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Prayagraj, Uttar Pradesh, India

**Mainak Bandyopadhyay** School of Computer Engineering, KIIT Deemed to Be University, Bhubneswar, India

**Sriparna Banerjee** ETCE Department, Jadavpur University, Kolkata, India

**Rakesh Baral** School of Computer Engineering, KIIT Deemed to Be University, Bhubneswar, India

**Rajni Bedi** Department of Computer Science Engineering, Lyallpur Khalsa College Technical Campus, Jalandhar, Punjab, India

**H. S. Behera** Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Sambalpur, Odisha, India

**Krishnendu Bera** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**S. Bharti** School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India

**Saranya Bhattacharjee** Department of Computer Science and Engineering, University of Engineering and Management, Kolkata, West Bengal, India

**Geetanjali Bhoi** Department of Computer Application, Veer Surendra Sai University of Technology, Burla, Odisha, India

**Samit Biswas** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India

**Susanta Chakraborty** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India

**Sanjana Chakravarty** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**Saurabh Chandra** CMR Institute of Technology, Bengaluru, India

**D. Chandrasekhar Rao** Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Sambalpur, Odisha, India

**Amitava Chatterjee** Electrical Engineering Department, Jadavpur University, Kolkata, India

**Sankhadeep Chatterjee** Department of Computer Science and Technology, University of Engineering and Management, Kolkata, West Bengal, India

**Krittika Chaturvedi** School of Information Technology and Engineering, VIT, Vellore, India

**Swati Chowdhuri** EEE Department, Institute of Engineering and Management, Kolkata, India

**C. Christlin Shanuja** CMR Institute of Technology, Bengaluru, India

**Simone Cuconato** Department of Informatics, Modeling, Electronics and Systems Engineering, University of Calabria, Cosenza, Italy;  
Institute of Informatics and Telematics (IIT)-CNR, Pisa-Cosenza, Italy

**Sanhita Dan** Department of CST, IIEST, Howrah, West Bengal, India

**Esther Daniel** Karunya Institute of Technology and Science, Coimbatore, India

**Ankur Das** Department of Software Engineering, Concordia University, Montreal, Canada

**Asit Kumar Das** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India

**Debashis Das** Department of Computer Science and Engineering, Techno India University, Saltlake, West Bengal, India

**Himansu Das** School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India

**Khakon Das** Department of Electronics and Communication, University of Allahabad, Allahabad, Uttar Pradesh, India

**Saumendra Das** School of Management Studies, GIET University, Gunupur, India

**Shuvrajeet Das** Department of CSE, NIT Agartala, Paschim Barjalai, India

**Susmita Das** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India

**Alina Dash** Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, India

**Pandit Byomakesha Dash** Department of Information Technology, Aditya Institute of Technology and Management, Tekkali, K Kotturu, Srikakulam, Andhra Pradesh, India

**Pubali De** Electrical Engineering Department, Techno India Group, Kolkata, India

**Francesco De Pascale** Department of Communication Sciences, University of Teramo, Teramo, Italy

**Ankit Deb** Department of CSE, NIT Agartala, Paschim Barjali, India

**Suman Deb** Department of Computer Science and Engineering, National Institute of Technology, Agartala, Barjali, Jirania, India

**Shantonu Debnath** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India

**Seema Dewangan** Department of Computer Science and Information Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur, Chhattisgarh, India

**R. Dhanushraj** Department of ECE, Mepco Schlenk Engineering College, Sivakasi, India

**Vikram Dhiman** Department of Computer Science Engineering, Gandhi Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India

**Arpita Dutta** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India

**Sangita Dutta** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India

**Debadatta Amaresh Gadanayak** Department of Electrical and Electronics Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar, India

**Dhruv Gandhi** Department of Computer Engineering, SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

**Arpan Garai** Department of CSE, IIT, Delhi, India

**Ahona Ghosh** Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, India

**Rajdeep Ghosh** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**Tulasichandra Sekhar Goripotu** Department of Electrical and Electronics Engineering, Sri Sivani College of Engineering, Srikakulam, Andhra Pradesh, India

**A. Gupta** School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India

**Priti Halder** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India

**Kevin Haria** Department of Computer Engineering, SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

**S. Hariharan** Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, India

**Anuja Jadhav** Department of Information Technology, Pimpri Chinchwad College of Engineering, Pune, India

**R. Jagadeeswara** Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, India

**Subit K. Jain** Department of Mathematics and Scientific Computing, National Institute of Technology Hamirpur, Himachal Pradesh, India

**Basra Jehangir** Amity School of Engineering and Technology, Amity University, Noida, India

**Jyoti** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Howrah, India

**A. Kala** Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India

**Tejas Kesarwani** Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India

**Y. Khandelwal** School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India

**Ashish Khare** Department of Electronics and Communication, University of Allahabad, Allahabad, Uttar Pradesh, India

**S. Khatri** School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India

**Ch Ravi Kishore** Department of Computer Science and Engineering, Aditya Institute of Technology and Management, Tekkali, K Kotturu, Srikakulam, Andhra Pradesh, India

**Venkatasai Kommu** Department of Computer Science and Engineering, Aditya Institute of Technology and Management, Tekkali, K Kotturu, Srikakulam, Andhra Pradesh, India

**Richa Kumari Kora** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Howrah, India

**Kannan Krishivasan** School of Computing, SASTRA Deemed University, Thanjavur, India

**Malay Kule** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**Ankit Kumar** Department of Mathematics and Scientific Computing, National Institute of Technology Hamirpur, Himachal Pradesh, India

**K. V. S. S. Kumar** School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India

**S. N. Kumar** Department of EEE, Amal Jyothi College of Engineering, Kottayam, Kerala, India

**Santosh Kumar** Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India

**Sumit Kumar** Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh, India

**V. Kumar** School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India

**Diptadip Maiti** Department of Computer Science and Engineering, Techno India University, Saltlake, West Bengal, India

**Soumyajit Maity** Department of Computer Science and Engineering, University of Engineering and Management, Kolkata, India

**Mishi Makade** Department of Computer Science and Information Technology, Sol Plaatje University, Kimberley, South Africa

**Ankan Mallick** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India

**Siddhartha Mallick** Indian Institute of Engineering Science and Technology, Shibpur, India

**P. Menaga** Department of Computer Science and Business Systems, Rajalakshmi Engineering College, Tamil Nadu, Chennai, India

**Manohar Mishra** Department of Electrical and Electronics Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar, India

**Subhasree Mohapatra** Institute of Technical Education and Research, Sikhsa ‘O’ Anusandhan University, Bhubaneswar, Odisha, India

**Ayatullah Faruk Mollah** Department of Computer Science and Engineering, Aliah University, Kolkata, India

**Supriya Mondal** EEE Department, Institute of Engineering and Management, Kolkata, India

**Swarnali Mondal** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**Abhiroop Mukherjee** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**Pratyush Kumar Muni** Department of Electrical and Electronics Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar, India

**Bighnaraj Naik** Department of Computer Application, Veer Surendra Sai University of Technology, Burla, Odisha, India

**Kshiramani Naik** Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Odisha, India

**Mehwish Naushin** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India

**Janmenjoy Nayak** Department of Computer Science, Maharaja Sriram Chandra Bhanja Deo University, Baripada, Odisha, India

**Soumya Ranjan Nayak** School of Computer Engineering, KIIT Deemed to Be University, Bhubaneswar, Odisha, India

**Devika Nikam** Department of Information Technology, Pimpri Chinchwad College of Engineering, Pune, India

**Ibidun Christiana Obagbuwa** Department of Computer Science and Information Technology, Sol Plaatje University, Kimberley, South Africa

**Etuari Oram** Department of Computer Application, Veer Surendra Sai University of Technology, Burla, Odisha, India

**Tannistha Pal** Department of Computer Science and Engineering, National Institute of Technology, Agartala, Barjala, Jirania, India

**Tamal Pal** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Howrah, India

**Asish Kumar Panigrahi** Department of Electrical and Electronics Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha, India

**Tanmoy Parida** Department of Electrical and Electronics Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar, India

**Renuka Devi Parlapalli** Mother Theressa College of Engineering and Technology, Peddapally, Telangana, India

**Nilesh Patil** Department of Computer Engineering, SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

**Chinmoy Kumar Patra** Department of Electrical and Electronics Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar, India

**Jayanta Paul** Indian Institute of Engineering Science and Technology, Shibpur, India

**Suresh Kumar Pemmada** Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University), Visakhapatnam, India

**Livio Perra** Dipartimento di Storia, Scienze dell’Uomo e della Formazione, Università degli Studi di Sassari, Sassari, Italy

**Teresa Perra** Azienda Ospedaliero Universitaria di Sassari, Sassari, Italy

**Alberto Porcu** Azienda Ospedaliero Universitaria di Sassari, Sassari, Italy

**Pranay Prajapati** Department of Computer Engineering, SVKM’s Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

**S. Pravinraj** Incedo Inc., Chennai, India

**Mauro Preda** Adjunct Professor (2000–2015), Università Cattolica of Milan, Milan, Italy

**R. Ahila Priyadharshini** Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

**S. Priyanga** Department of Computer Science and Engineering, Srinivasa Ramanujan Centre, SASTRA Deemed University, Kumbakonam, India

**Saurabh Raj** CMR Institute of Technology, Bengaluru, India

**V. Rajalakshmi** Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India

**Divya Rajput** Department of Information Technology, Pimpri Chinchwad College of Engineering, Pune, India

**Vysyaraju Lokesh Raju** Tata Consultancy Services Pvt Ltd, Pune, Maharashtra, India

**Anjan Rakshit** Electrical Engineering Department, Jadavpur University, Kolkata, India

**K. Ramajeyam** Department of ECE, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

**Hiral Rane** Department of Information Technology, Pimpri Chinchwad College of Engineering, Pune, India

**G. V. Eswara Rao** Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Prayagraj, Uttar Pradesh, India

**Rajwant Singh Rao** Department of Computer Science and Information Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur, Chhattisgarh, India

**Ch. Ravi Kishore** Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Sambalpur, Odisha, India

**Venkata Bhavana Repalle** College of Business, The University of Tampa, Tampa, USA

**Joan Rosselló-Geli** Universitat Oberta de Catalunya, Barcelona, Spain

**Anuska Roy** Indian Institute of Engineering Science and Technology, Shibpur, India

**Gaetano Sabato** Department of Psychological, Pedagogical, Physical Exercise and Training Sciences, University of Palermo, Palermo, Italy

**Sriparna Saha** Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, India

**Taniya Sahana** Department of Computer Science and Engineering, Dumkal Institute of Engineering and Technology, Basantapur, Murshidabad, India;  
Department of Computer Science and Engineering, Aliah University, Kolkata, India

**Anurag Sahu** Department of Biotechnology and Bioinformatics, Vignan Foundation For Science Technology and Research, Vadlamudi, Andhra Pradesh, India

**Rabindra Kumar Sahu** Department of Electrical and Electronics Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha, India

**Piyush Kanti Samanta** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India

**S. Seetha** CMR Institute of Technology, Bengaluru, India

**Arnab Sen** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**Manas Ranjan Senapati** Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Sambalpur, Odisha, India

**Hritik Shah** School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India

**V. S. Shankar Sriram** School of Computing, SASTRA Deemed University, Thanjavur, India

**Pallavi Sharma** Department of Electronics and Communication Engineering, Lyallpur Khalsa College Technical Campus, Jalandhar, Punjab, India

**P. Sharon Femi** Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, Tamil Nadu, India

**Subinoy Sikdar** Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India

**Jaya Sil** Indian Institute of Engineering Science and Technology, Shibpur, India

**Mansi Singh** School of Information Technology and Engineering, VIT, Vellore, India

**Navnee Singh** Moneyview, Bangalore, Karnataka, India

**Raunaq Singh** Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Prayagraj, Uttar Pradesh, India

**Vipin Singh** Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Prayagraj, Uttar Pradesh, India

**P. Sivaramapandian** Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

**S. Sivavakisan** Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

**E. Srikanth Reddy** Vaageswari College of Engineering, Karimnagar, Telangana, India

**Sharmila Subudhi** Department of Computer Science, Maharaja Sriram Chandra Bhanja Deo University, Baripada, Odisha, India

**H. Swapnarekha** Department of Information Technology, Aditya Institute of Technology and Management, Srikakulam, Andhra Pradesh, India

**Elisabeth Thomas** Lincoln University College, Kota Bharu, Malaysia

**B. K. Tripathy** School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India

**Pooja Verma** Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Prayagraj, Uttar Pradesh, India

**Shreya Verma** School of Information Technology and Engineering, VIT, Vellore, India

**M. Vijayraj** Department of ECE, Mepco Schlenk Engineering College, Sivakasi, India

**S. Vimal** Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India

**Janhavi Wagh** Department of Information Technology, Pimpri Chinchwad College of Engineering, Pune, India

**Shaiq Wani** Department of Computer Science Engineering, Chandigarh University, Gharaun, Mohali, India

**Vincent Mohale Zibi** Department of Computer Science and Information Technology, Sol Plaatje University, Kimberley, South Africa

# A New Technique of Cipher Type Identification Using Convolutional Neural Networks



Subinoy Sikdar and Malay Kule

**Abstract** This paper focuses on modern cipher types classification mechanism using convolutional neural networks (CNNs). In case of ciphertext-only-attack, it is momentous to a cryptanalyst to recognize the cipher type first for further cryptanalysis work, usually when the messages are transmitted from an unknown source. In this research paper, two modern ciphers named as AES-128 and RC-4 have been considered for the classification problem. At first, a corpus is created. Messages from the corpus are encrypted using these two encryption algorithms, and the ciphertexts are recorded into punched tape. We have taken the images of punched tapes and put those images into the dataset. These images are provided to different pretrained CNN architecture models to train the network models. For the training purpose, we have used transfer learning method. The results obtained from two different CNN architectures such as ResNet50V2 and MobileNetV2 have been compared and plotted. The primary success of this research is that a very much complex problem is solved in a much simpler way. The experimental results prove the validity of our proposed work.

**Keywords** Convolutional neural network · Ciphertext · Cryptanalysis · AES-128 · RC-4

## 1 Introduction

Information is the power in present days. Information and data need to be protected from unauthorized access, distortion or any kind of unauthorized activity. Cryptography [1] is the study of secret writing. Online communication takes place over the

---

S. Sikdar (✉) · M. Kule

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur 711103, India  
e-mail: [subinoysikdar.jume2019@gmail.com](mailto:subinoysikdar.jume2019@gmail.com)

M. Kule  
e-mail: [malay.kule@gmail.com](mailto:malay.kule@gmail.com)

network or in the communication channel. Due to presence of network adversaries over the network, information becomes no more secret. Our communication channel is unreliable; a third party can secretly peep over the channel and can get the information from the network or from the channel. Here, cryptography plays an important role to protect our data from unauthorized access. Cryptography maps original messages (plaintext) into unreadable format (ciphertext) using key(s) and the vice versa. The process of encoding original message into unreadable format using key(s) is versed as encryption, and the contrary method of decoding the ciphertext into plaintext using key(s) is known to decryption. Only the authorized user can encrypt or decrypt the messages.

Cryptanalysis [2] is the detailed study of a cryptosystem to find out the weakness, flaws of the system and aims for breaking the cryptographic systems by finding the key(s) or without key(s). So, cryptanalysis helps to get a robust cryptographic system by continuously examining the weakness and flaws of the system. Generally, every cryptosystem is breakable at some point of time with huge computation power and resources. But the limitation is time and resource power. A cryptanalyst has to break the cryptosystem within a limited time and limited resource and computation power. Cryptanalysis itself is a very much time-consuming task. From the past data, it has been seen that a very large amount of time is taken to detect the cipher type first. So, it would be very helpful if we know the type of the cipher beforehand, and it will lead us to apply the appropriate cryptanalysis method in a very reasonable time. It will definitely accelerate the cryptanalysis work.

In recent days, machine learning has a lot of influence over every field of computer science. Application of machine learning (ML) in cryptanalysis has made cryptanalysis work much easier for a complex cryptanalysis problem. There exist many real-time cryptographic problems which are solved with the help of artificial intelligence and machine learning; hence, these problems reside in the overlapping region of cryptanalysis and machine learning [3].

Various important research works on cipher type detection using multiple techniques have been reported in the recent past. Ahmadzadeh et al. [4] proposed a ciphertext categorization model subjected on recurrent neural network (RNN) using natural language processing (NLP). A bidirectional long short-term memory (LSTM) and gated recurrent units (GRU) network was used for this classification problem. Substitution cipher, Caesar cipher and Vigenère cipher were considered as three different class labels.

Abd et al. [5] introduced a classical ciphertext classification model using artificial neural network (ANN). The classifier was built based on a hierarchical manner with three different levels of classification followed by one by one. Seven statistical features were considered for the classification. They are DIU, DISU, DISB, Bigrams (BIG), Trigrams (TRI), Quadgrams (QUA) and Caesar (CAE).

Sharif et al. [6] proposed an identification scheme of different block ciphers using pattern recognition techniques. Four categories of block ciphers were chosen for this classification problem such as DES, IDEA, AES and RC. The encryption was done in Electronic Code Book (ECB) mode of operation. Eight different machine learning algorithms were employed to execute the differentiation problem. The algorithms are

Naïve Bayesian (NB), AdaBoost, support vector machine (SVM), random forest, instance-based learning (IBL), neural network (MLP), bagging (Ba) and decision tree (C4.5). From the experimental results, it was seen that random forest was the best classification algorithm for this ciphertext classification problem.

Chuxuan [7] proposed a cryptosystem recognition scheme depending on convolution attributes. The main focus of this paper was to pluck out different attributes from the ciphertext by linear transformation and convolution sampling. A classification model was built using random forest classification algorithm, and the model was trained with the extracted convolution features. In this research work, four different ciphertexts were considered like AES, 3DES, Blowfish and RSA as the class labels.

Leierzopf et al. [8] proposed a classical cipher type detection method using feature learning approach. They used two feature learning algorithms such as LSTM and transformer to overcome the dependencies on domain knowledge. A combination of feature learning and feature engineering in neural network achieved a good accuracy. Transfer learning technique was used while training the networks.

Kopal [9] proposed a ciphertext classification technique using artificial neural network using feature engineering and feature learning approach. The notable importance of his work was that the model could save 54% computation time in ciphertext detection rather than other classification algorithm. Five different classical cipher types were considered to this classification problem. The cipher types are Simple Monoalphabetic Substitution cipher, Playfair cipher, Vigenère cipher, Hill cipher and Transposition cipher.

In this research work, we have proposed a technique for cipher types identification using CNNs. First of all, a corpus is created which contains single character; different symbols; small sentences of two, three or four words; English pangram sentences; English poems; and some short stories of English. All these data from the corpus are taken and encrypted using both the encryption algorithm AES-128 and RC-4. These ciphertexts are recorded in the punched tape, and we have taken images of the punched tape to train deep neural network. After the training of these CNN models, it acquired a very good accuracy in both training and testing accuracy. The models are also tested manually with satisfactory predictions of cipher types.

This paper is assembled in six different sections. Section 1 inaugurates the paper. Section 2 talks about the objective and the motivation of this project. Section 3 describes the preliminaries of this paper followed by a detailed discussion, implementation of proposed method of cipher types detection using convolutional neural networks algorithm has been mentioned in Sect. 4, and Sect. 5 analyzes the outcomes obtained from the experiment mentioned in Sect. 4. Finally, Sect. 6 directs the future scopes and conclusion of this paper.

## 2 Objective and Motivation

Our first and foremost interest behind this research work was to reduce the computation time required for cryptanalysis. It is thought in general that the encryption algorithm should always be public. The only security lies inside an encryption algorithm is in its secret keys. But in a real-time scenario, this is not always the case that the encryption algorithm is always public. In the real-time scenario, the only available thing in our hand may be the ciphertext. In case of ciphertext-only-attack (COA), knowing the cipher type beforehand plays a major role for further cryptanalysis work. Nowadays, there are several examples where the messages are transmitted from an unknown suspicious source, and we really do not have any idea about the category of the cipher in which it belongs to. In recent days, several terrorist activities are happening near the border line areas of the countries to execute security attacks. Intelligence departments always try to cryptanalyze and decode this kind of unauthorized communication to prevent the terrorist attacks. Once the cipher type is identified, we can start cryptanalysis using standard cryptanalysis methods applicable on different ciphers. For example, linear cryptanalysis and differential cryptanalysis were introduced for cryptanalysis of block cipher, whereas Side-Channel-Analysis-Attack, Divide-and-Conquer-Attack, Algebraic-Attack and Exhaustive-Key-Search-Attack are the well-known cryptanalysis techniques exist for stream cipher cryptanalysis.

Ciphertext feature extraction is very important to decode a ciphertext. Each encryption algorithm follows certain mathematical and logical rules. So, the ciphertexts are also generated as per the inherent rules implemented within the encryption algorithms. Identifying certain patterns helps in decoding the ciphertext. Expect some well-known online platform; we really are not aware about encryption algorithms. In case of an unknown source, the encryption algorithm remains hidden, specially, in case of border side terrorist activity. Our objective is to make an automated computer-aided design (CAD) tool which can detect the cipher type and help the cryptanalyst to perform further cryptanalysis work with appropriate applicable cryptanalysis technique in a limited time. So, determining the cipher type by our proposed model will save a lot of time for the cryptanalyst. In this paper, we have considered two modern cipher types such as AES-128 and RC-4 in apropos to test the correctness of our proposed technique. Any other stream/block cipher algorithms can also be used for the same.

## 3 Preliminaries

In this paper, two different class of modern ciphers have been considered while conducting the experiment. They are AES-128 and RC-4. Both the ciphers are used in recent days for secure communications. We have chosen these ciphers because in block cipher AES is the highly popular block cipher and RC is highly popular in

stream cipher used by several popular organizations. These ciphers are discussed in brief as follows.

### 3.1 AES-128 Cipher

AES [10] stands for Advance Encryption Standard. It is a 128-bit block cipher with 10 round keys. AES [11] encryption algorithm consists of main four operations such as add Round Key, S Byte or S Box operation, Shift Row and Shift Column operation. Mix column is omitted in the last round. AES-128 encipher process is briefly described below [12].

- **Round Key Operation**—Add Round Key Operation is just bitwise XOR operation in between plaintext bits and key bits ( $p_i \oplus k_i$ ).
- **Sub Byte/S Box Operation**—S Box will take an 8-bit input and produce an 8-bit output.
- **Shift Row Operation**—In the state matrix, some state in the row will interchange their positions.
- **Mix Column Operation**—In the state matrix, each column will be converted into a new column.

### 3.2 RC-4 Cipher

RC-4 is modern symmetric key stream cipher [13]. It encodes a byte at once. There are mainly four processes involved in RC-4 encryption process. They are S-array state initialization, key scheduling algorithm, pseudo-random key stream generation and XOR operation in between plaintext and key stream bits.

- **S-array/S-matrix**

$$[S_0, S_1, S_2, S_3, \dots, S_{255}] = [0, 1, 2, 3, \dots, 255]$$

$$\text{Key} = [\kappa_0, \kappa_1, \kappa_2, \kappa_3, \dots, \kappa_{n-1}]$$

$$\text{Key-state-array} = [K_0, K_1, K_2, K_3, \dots, K_{255}]$$

$$\text{Key-array} = [\kappa_0, \kappa_1, \kappa_2, \dots, \kappa_{n-1}, \kappa_0, \kappa_1, \kappa_2, \dots, \kappa_{n-1}, \kappa_0, \kappa_1, \kappa_2, \dots, \kappa_{n-1}, \dots]$$

$$\text{Plaintext} = [p_0, p_1, p_2, p_3, \dots, p_{n-1}]$$

- ***Key Scheduling Algorithm***

The initialized ‘S’ array is run through key scheduling algorithm (KSA). Key scheduling algorithm uses the secret key to scramble the S-array.

- ***Pseudo-random Stream Generation***

The actual key stream is generated by the pseudo-random stream generation (PRSG) function with KSA scrambled ‘S’ array.

- ***XOR Operation***

This is the final step, where ciphertext is generated by performing bitwise XOR in between plaintext bit and key stream bit.

$$\text{ciphertext} = [p_i \oplus x_i]; i = 0, 1, 2, \dots, n - 1$$

### 3.3 Cryptanalysis

Cryptanalysis [1] is the detailed study of a cryptosystem to find out its weakness and flaws in order to get a more improved strongly secure cryptosystem. There are several classes of cryptanalysis techniques available in the present time such as frequency analysis, brute-force-attack, ciphertext-only-attack, known-plaintext-attack, chosen-plaintext-attack, chosen-ciphertext-attack, linear cryptanalysis, side-channel-analysis, differential cryptanalysis, fault analysis, correlation power analysis, differential power analysis, man in the middle attack and many more.

### 3.4 Convolutional Neural Network (CNN)

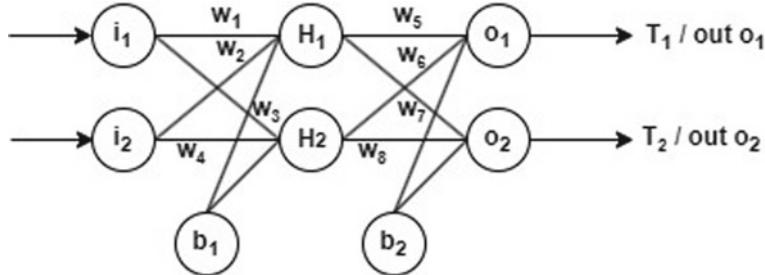
Convolutional neural networks (ConvNets) [14] are basically an image processing tool. When images are passed through a ConvNet, they go through various layers. The layers are convolution, pooling, flattening and full connection layer.

- ***Convolutional Layer***

A convolution [15] describes the region of imbricate of one function when it is permuted over another function. In image processing, convolution operations means that different feature detectors are passed over the images to obtain different feature maps. Along with convolution, a rectified linear unit (ReLU) is used to raise nonlinearity inside the images.

- ***Pooling Layer***

Pooling handles distorted images like stretched, compressed, rotated, etc. Pooling helped to preserve the important features in a distorted image, and it also reduces the



**Fig. 1** Artificial neural network (ANN)

image size by 75%. It is also a filter which passes over the images and extracts the pooled feature maps from the images.

- **Flattening**

This is a very simple and quick step. In this step, the pooled feature maps are converted into a one-dimensional array which will be fed to fully connected layer for further processing.

- **Fully Connected Layer**

This is the final step of ConvNets which are none other than artificial neural network (ANN) [14] described in Fig. 1. Input neurons are fed inputs from the input feature vector and passed through several neurons to predict the output. These predicted output values are compared with the actual outputs, resulting the error in the output. This error will be backpropagated in the network to adjust the weights. Gradient descent function is used to determine the error in the weights.

### Forward Propagation

It is calculated using Eq. 1.

$$H_1 = i_1 w_1 + i_2 w_2 + b_1; \quad H_2 = i_1 w_3 + i_2 w_4 + b_2 \quad (1)$$

### Activation Function

$$\text{out}H_1 = \frac{1}{1 + e^{-H_1}} \quad (2)$$

Similarly,  $\text{out}H_2$ ,  $\text{out}o_1$  and  $\text{out}o_2$  will be calculated using Eq. 2.

**Total Error:** Error is calculated using the following Eq. 3:

$$E_{\text{Total}} = \sum \frac{1}{2} (\text{target} - \text{output})^2 \quad (3)$$

**Backpropagation:** Updating the weights.

Consider the weight  $w_5$ : Error at  $w_5$  is shown in Eq. 4.

$$E_{w_5} = \frac{\delta E_{\text{Total}}}{\delta w_5} \quad (4)$$

Updating the weight  $w_5$  is done as shown in Eq. 5.

$$w_5 = w_5 - \lambda \frac{\delta E_T}{\delta w_5} \quad (5)$$

Here,  $\lambda$  denotes learning rate ( $0 < \lambda < 1$ ).

Similarly,  $w_1, w_2, w_3, w_4, w_6, w_7$  and  $w_8$  will be updated using the same chain rule, and the process of upgradation will be continued until the error is minimized.

### 3.5 Punched Tape

Punched tapes [16–18] were used in early days to store input data in a computer. Instead of punch cards, a punched tape is a long strip which can be rolled and store more information by containing punched holes on the tape that represents the data being 0 or 1. An example of punched tape encoding scheme is illustrated below in the Table 1.

## 4 Proposed Method of Cipher Type Identification

Our proposed method of cipher types detection process is divided into four modules such are as Corpus Creation, Encryption Process, Dataset Generation and Building Classifier Model. Each module is explained below.

**Table 1** Punched tape encoding scheme

Text	Baudot code	Punched tape	Bytes	
			Binary	Group by 5 bits
SUBINOY	5 bytes encoded		00101 11001 01100 10101	00111 00110 11000 00000

#### **4.1 Corpus Creation**

First of all, a corpus is created. The corpus contains famous drama written by W. Shakespeare named as Macbeth.

#### **4.2 Encryption Process**

In this research paper, AES-128 and RC-4 ciphers have been considered for the cipher types classification problem. The messages are enciphered using these two encoding methods. All the messages from corpus are encrypted and recorded into a punched tape. We have taken a snapshot of the punched tape and stored the images into the dataset.

- **RC-4 Encryption Process**

**Key:** 63 72 79 70 74 69 69

**Punched Tape Row Size:** 20

- **AES-128 Encryption Process**

**Mode:** Cipher Block Chaining (CBC)

**Key:** 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

**Initialization Vector:** 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f

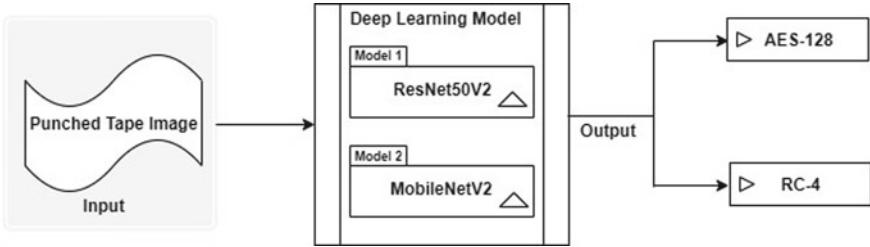
**Punched Tape Row Size:** 20

#### **4.3 Dataset Generation**

Here, the dataset is the collection of all the encrypted punched tape images. Dataset contains 2000 images. We have divided our dataset into three folders such as Training\_Set, Test\_Set and Validation\_Set. All these three folders contain two subfolders such as AES-128 and RC-4. Inside the Training\_Set, both the AES-128 and RC-4 folders are filled with 800 sample images in each folder. Similarly, in the Test\_Set, both the folders contain 200 sample images each. In the Validation\_Set, we have kept 100 AES-128 cipher images and 100 RC-4 cipher images.

#### **4.4 Building the Classifier Model**

- **Experimental Setup**—This experiment was carried out on Google Colab environment with a personal computer.
- **Classifier Model**—This classification problem was solved with the help of deep learning, and we have used transfer learning technique and build deep neural



**Fig. 2** Implemented deep learning model

networks (DNN). All the necessary libraries under the Keras module are imported such as Input, Lambda, Dense, Flatten and ImageDataGenerator. Image size is provided with the dimension 224, 224 with RGB channels. Training and validation path is defined. We have used a generic code so that we can employ our desired deep learning network by mentioning their names. In accordance with, ResNet50V2 [15] and MobileNetV2 [19] are imported with the existing weights of IMAGENET. The first and the last layers of the two deep neural networks are discarded and made a flatten layer (input layer) and the last layer (output layer) with the length of the dataset. Here, the length of our dataset is 2 (AES-128, RC-4). In the last layer, softmax activation function was used. Finally, the input and output predictions were combined and developed the classifier model. The model was compiled with ‘adam’ optimizer, ‘accuracy’ matrix and the loss function as ‘categorical cross\_entropy’. Dataset was uploaded using the Keras module with ImageDataGenerator library and also created some more additional dataset like rescaling, zoom ranging, horizontal flipping, etc. After that, images were inserted using the library flow\_from\_directory from the training set as well as from the test set for the validation purpose with target size 224, 224; batch size 32 and the class mode were categorical. Finally, the fit\_generator was called with 20 epochs, and our model was trained. After the training was done, we saved the model as.h5 file and made a framework where we can provide the cipher image, and our model predicts the cipher type with a very high accuracy. Hence, our model was validated. Figure 2 shows the implemented deep learning [14, 20] model for cipher types detection problem.

## 5 Experimental Results and Analysis

Dataset created by us contains 1000 samples each from AES-128 and RC-4. Dataset is organized into three subfolders such as training set, test set and validation set. About 20% data are kept in the test set, and the rest is in the training set. Two different CNN architectures such as ResNet50V2 and MobileNetV2 were used to solve this problem. All the results are compared and showed by graphical representation. All the CNN architectures have successfully classified the AES-128 and RC-4 cipher.

Both the models ReNet50V2 and MobileNetV2 have been considered to be the deployed model as it gave the highest accuracy for the classification problem.

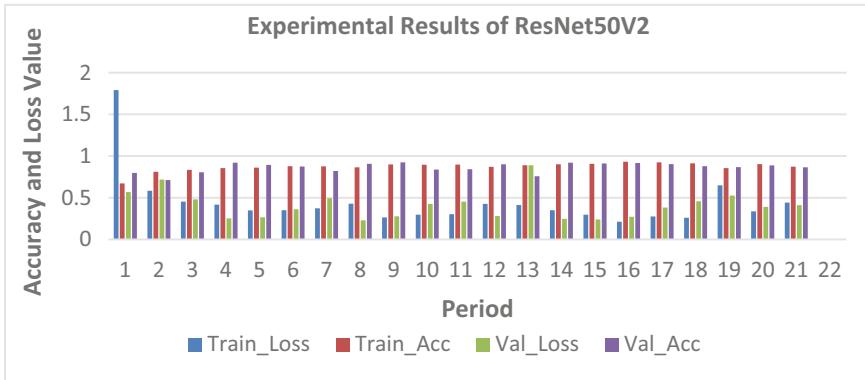
In this section, we have discussed the empirical outcomes obtained while validating the models.

## 5.1 ResNet50V2

Table 2 shows experimental results of ResNet50V2. We have obtained average training loss 0.44117 and average training accuracy 0.87192; average validation loss 0.41018 and average validation accuracy 0.8639 from ResNet50V2. There was 200,706 number of trainable parameters in ResNet50V2. In Fig. 3, in the epoch axis, 21st epoch denotes the average loss and accuracy obtained from ResNet50V2 in 20 epochs.

**Table 2** Results obtained from ResNet50V2

Period	Train_Loss	Train_Acc	Val_Loss	Val_Acc
1	1.7912	0.6705	0.5679	0.7980
2	0.5843	0.8111	0.7179	0.7119
3	0.4533	0.8336	0.4800	0.8046
4	0.4170	0.8544	0.2524	0.9205
5	0.3471	0.8602	0.2648	0.8940
6	0.3497	0.8785	0.3615	0.8742
7	0.3732	0.8769	0.4936	0.8212
8	0.4278	0.8644	0.2291	0.9073
9	0.2634	0.8993	0.2769	0.9238
10	0.2964	0.8960	0.4242	0.8377
11	0.3012	0.8977	0.4536	0.8411
12	0.4265	0.8702	0.2802	0.9007
13	0.4130	0.8902	0.8889	0.7583
14	0.3498	0.9010	0.2458	0.9205
15	0.2968	0.9077	0.2389	0.9106
16	0.2111	0.9326	0.2721	0.9172
17	0.2749	0.9235	0.3823	0.9040
18	0.2605	0.9118	0.4577	0.8775
19	0.6490	0.8561	0.5256	0.8675
20	0.3372	0.9027	0.3902	0.8874
Average	0.44117	0.87192	0.41018	0.8639



**Fig. 3** Training loss versus validation loss and training accuracy versus validation accuracy (ResNet50V2)

## 5.2 MobileNetV2

Table 3 shows experimental results of MobileNetV2. We have obtained average training loss 0.765125 and average training accuracy 0.832975; average validation loss 0.66364 and average validation accuracy 0.822675 from MobileNetV2. There was 125,442 number of trainable parameters available in MobileNetV2.

Figure 4 depicts the training loss vs. validation loss plot in 20 epochs and also the training accuracy and the validation accuracy on the same number of epochs.

Figure 4 shows that, in case of MobileNetV2, the accuracy reached in both the case of training and validation is comparatively lower than ResNet50V2. In case of loss, both the training and validation losses are comparatively higher than ResNet50V2. From this point of view, ResNet50V2 can perform better. But we do not need to worry as both the models have performed well in a real-time system while manual testing.

## 5.3 Comparison Between Proposed Models

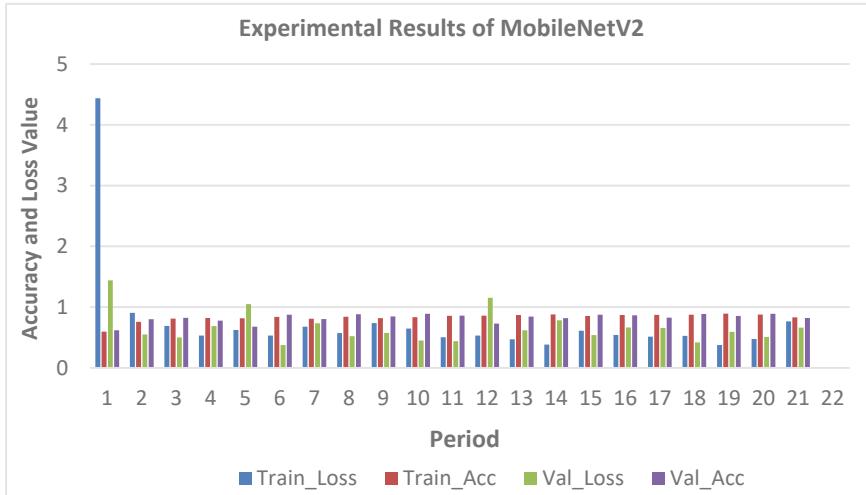
In this module, we have shown the average experimental results obtained from each epoch and put the results in a tabular format. Table 4 shows the average results of ResNet50V2 and MobileNetV2.

Figure 5 presents the comparison of average results of ResNet50V2 and MobileNetV2. Considering the obtained results, it is clear that ResNet50V2 has gained higher accuracy over MobileNetV2. But in real time, while validating (manual testing) the models, both the models have performed very well and classify the cipher types accurately. So, both the models can be employed in the real-time systems. These

**Table 3** Results obtained from MobileNetV2

Period	Train_Loss	Train_Acc	Val_Loss	Val_Acc
1	4.4375	0.5957	1.4398	0.6192
2	0.9071	0.7579	0.5497	0.8013
3	0.6921	0.8111	0.5022	0.8245
4	0.5322	0.8228	0.6880	0.7781
5	0.6255	0.8153	1.0498	0.6788
6	0.5322	0.8386	0.3754	0.8742
7	0.6776	0.8087	0.7356	0.8046
8	0.5723	0.8419	0.5218	0.8841
9	0.7362	0.8186	0.5764	0.8477
10	0.6483	0.8336	0.4515	0.8907
11	0.5055	0.8569	0.4397	0.8609
12	0.5317	0.8602	1.1546	0.7285
13	0.4706	0.8702	0.6201	0.8444
14	0.3845	0.8810	0.7829	0.8179
15	0.6125	0.8561	0.5393	0.8742
16	0.5420	0.8710	0.6650	0.8642
17	0.5144	0.8727	0.6589	0.8278
18	0.5283	0.8760	0.4193	0.8874
19	0.3755	0.8935	0.5942	0.8543
20	0.4765	0.8777	0.5086	0.8907
Average	0.765125	0.832975	0.66364	0.822675

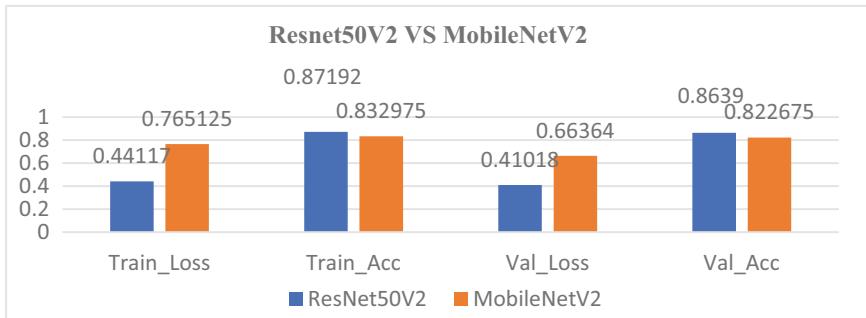
models are really helpful in identifying the cipher types accurately with in a minimal time.



**Fig. 4** Training loss versus validation loss and training accuracy versus validation accuracy (MobileNetV2)

**Table 4** Average results of ResNet50V2 and MobileNetV2

CNN_Model	Train_Loss	Train_Acc	Val_Loss	Val_Acc
ResNet50V2	0.44117	0.87192	0.41018	0.8639
MobileNetV2	0.765125	0.832975	0.66364	0.822675



**Fig. 5** Comparison of average results of ResNet50V2 and MobileNetV2

## 6 Conclusion

This paper demonstrates the modern cipher types classification problem. We have used the punched tape images of the encoded ciphers and put these images into our models as input, and our model can successfully predict the cipher types accurately.

Though we have trained our models to differentiate the two modern ciphers AES-128 and RC-4, it has been observed that our models can also differentiate between block cipher and stream cipher. Even while testing, we provide some variation of block cipher (like AES-256); the models are able to predict that correctly. Moreover, the same is successfully done with the stream cipher also with a little variation. So, this is an extra fruitful outcome of our experiment. This paper will definitely help the cryptanalyst to do cryptanalysis work in a reasonable time. This work may further be extended to improve our models so that it can classify a long ranges of cipher types.

## References

1. Stallings W *Cryptography and network security: principles and practice*, Pearson
2. Forouzan BA (2007) *Cryptography and network security*. Tata McGraw-Hill, New Delhi
3. Khan AN, Yu Fan M, Malik A, Husain MA (2019) Cryptanalyzing Merkle-Hellman public key cryptosystem with artificial neural networks. In: 2019 IEEE 5th International conference for convergence in technology (I2CT), pp 1–7
4. Ahmadzadeh E, Kim H, Jeong O, Kim N, Moon I (2022) A deep bidirectional LSTM-GRU network model for automated ciphertext classification. *IEEE Access* 10:3228–3237
5. Abd AJ, Al-Janabi S (2019) Classification and identification of classical cipher type using artificial neural networks. *J Eng Appl Sci* 14(11):3549–3556
6. Sharif SO, Kuncheva LI, Mansoor SP (2010) Classifying encryption algorithms using pattern recognition techniques. In: 2010 IEEE International conference on information theory and information security, pp. 1168–1172
7. Chuxuan Y (2021) Cryptosystem recognition scheme based on convolution features. In: 2021 International conference on artificial intelligence, big data and algorithms (CAIBDA), pp 229–232
8. Leierzopf E, Mikhalev V, Kopal N, Esslinger B, Lampesberger H, Hermann E (2021) Detection of classical cipher types with feature-learning approaches. In: Data Mining. AusDM 2021. Communications in computer and information science, vol 1504. Springer, Singapore
9. Kopal N (2020) Of ciphers and neurons—detecting the type of ciphers using artificial neural networks. In: Proceedings of the 3rd International conference on historical cryptology HistoCrypt, no 171, pp 77–86
10. Heron S (2009) Advanced Encryption Standard (AES), Network Security, vol 2009, Issue 12, pp 8–12. ISSN 1353-4858
11. Kahate A *Cryptography and network security*. McGraw Hill Education (India) Private Limited
12. Bulens P, Standaert FX, Quisquater JJ, Pellegrin P, Rouvroy G (2008) Implementation of the AES-128 on Virtex-5 FPGAs. In: Vaudenay S (eds) *Progress in cryptology—AFRICACRYPT 2008*. AFRICACRYPT 2008. Lecture notes in computer science, vol 5023. Springer, Berlin, Heidelberg
13. Mantin I, Shamir A (2002) A practical attack on broadcast RC4. In: Matsui M (eds) *Fast software encryption*. FSE 2001. Lecture notes in computer science, vol 2355. Springer, Berlin, Heidelberg
14. Nielsen MA (2015) *Neural networks and deep learning*, vol 25. Determination Press, San Francisco, CA, USA
15. Albawi S, Mohammed TA, Al-Zawi S (2017) Understanding of a convolutional neural network. In: 2017 International conference on engineering and technology (ICET), pp 1–6
16. Maxfield C (2011) How it was: paper tapes and punched cards. *EE Times*
17. National Security Agency Central Security Service. Tale of the Tape. 3 May 2016. Retrieved 16 June 2014.

18. Shannon CE (1949) Communication theory of secrecy systems. *Bell Syst Tech J* 28(4):656–715
19. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on computer vision and pattern recognition, pp 4510–4520
20. Suba N, Verma A, Baviskar P, Varma S (2022) Violence detection for surveillance systems using lightweight CNN models. In: 7th International conference on computing in engineering and technology (ICCET 2022), pp 23–29

# Monthly Rainfall Forecasting Using Sequential Models



A. Kala, P. Sharon Femi, V. Rajalakshmi, and K. Ashwini

**Abstract** Weather forecasts especially rainfall forecasts, constitute complex tasks because they rely heavily on various parameters like humidity, wind direction, and wind speed, which change dynamically based on the climatic conditions. Accurate forecasts of rainfall are considered important in agriculture and other industries. Therefore, there is a great need for accurate rainfall prediction, and it can be achieved by deep learning algorithms. The challenging part lies in reducing the forecasting error rate. To reduce the forecasting error, this paper employs Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) sequential models to forecast monthly rainfall, and uses long-time series data. The dataset utilized for forecasting monthly rainfall comprises time series data spanning from 1871 to 2016 and encompasses the entirety of all India's monthly rainfall. It is obvious from the results that both RNN and LSTM models show better performance for reducing the forecasting error on monthly rainfall data. When evaluating the performance of these algorithms, the LSTM algorithm provides a good model for rainfall prediction.

**Keywords** Rainfall forecasting · Sequential models · RNN · LSTM

---

A. Kala (✉) · P. Sharon Femi

Department of Information Technology, Sri Venkateswara College of Engineering,  
Sriperumbudur, Tamil Nadu 602105, India

e-mail: [akala@svce.ac.in](mailto:akala@svce.ac.in)

P. Sharon Femi

e-mail: [sharon@svce.ac.in](mailto:sharon@svce.ac.in)

V. Rajalakshmi

Department of Computer Science and Engineering, Sri Venkateswara College of Engineering,  
Sriperumbudur, Tamil Nadu 602105, India

e-mail: [vraji@svce.ac.in](mailto:vraji@svce.ac.in)

K. Ashwini

Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa  
Vidyapeetham, Chennai 601103, India

e-mail: [k\\_ashwini@ch.amrita.edu](mailto:k_ashwini@ch.amrita.edu)

## 1 Introduction

Accurate prediction of meteorological time series statistics can be highly significant, especially for countries such as India where the economy is heavily reliant on agriculture. Therefore, any changes in annual rainfall impacts the agriculture in India. Having prior knowledge of monsoon behavior can assist Indian farmers and the government in making crucial decisions. It reduces the damage to crops during the monsoon season when rainfall is low. However, due to the nonlinear behavior of the climate, it is difficult to forecast the rainfall.

Researchers conducted various experiments to identify the predictive parameters suitable for forecasting all India monthly rainfall. Many historical methods were initially used, which only considered past data under the same circumstances and restricted their predictions to specific locations. Subsequently, multiple statistical techniques were developed to predict monthly rainfall, employing advanced statistical calculations that outperform traditional methods. However, these mathematical and statistical models necessitate extensive computational resources. In recent times, with the remarkable advancements in pattern recognition, a majority of researchers resort to deep learning models to examine precipitation patterns.

The paper is organized into several sections. In Sect. 2, the contributions of various researchers are examined. Section 3 examines the techniques and dataset used, while Sect. 4 assesses the effectiveness of the model. Finally, Sect. 5 presents the paper's conclusion.

## 2 Related Works

Chattpadhyay and Chattpadhyay [1] formulated a feed-forward ANN model to forecast mean rainfall during summer monsoon. They constructed a three-layer neural network with sigmoid nonlinearity to develop the predictive model. The input matrix for the ANN was generated by utilizing total monthly monsoon rainfall during summer, tropical rainfall index, and sea surface temperature anomalies as predictors. The neural network was compared with multiple linear regression, and other established ANN models.

Chattpadhyay and Chattpadhyay [2] established ANN model to forecast monsoon rainfall of India. They developed 19 neural network models with variable size hidden layers. After comprehensive training and testing procedures, using 11 nodes in hidden layer of neural network is most appropriate for predicting summer monsoon.

To forecast monthly rainfall in homogeneous regions of India, Kashid, and Maity [3] employed genetic programming as an artificial intelligence tool to observe the relationship between monsoon precipitation and atmospheric circulation indexes. According to their research, LSTM model exhibits superior performance than genetic programming model in forecasting monthly rainfall across India.

Bianchi et al. [4] applied Recurrent Neural Networks to predict time series data focusing on short-term load forecasting. They evaluated five distinct architectures, including STM, GRU, ESN, NARX, and ERNN, and analyzed their respective prediction performance on various time series. LSTM and GRU were deemed effective at modeling highly nonlinear statistical dependencies due to their gating mechanisms, which facilitate rapid modification of the cell's memory content and internal dynamics. They concluded that Recurrent Neural Networks and their variations could be effectively applied for time series prediction.

Singh [5] introduced a novel technique for forecasting Indian Summer Monsoon Rainfall (ISMR) by combining ANN, fuzzy sets, and entropy. Time series datasets were utilized to forecast ISMR for monthly and seasonal timescales. Fuzzy sets were employed to represent the uncertainty associated with ISMR values, while entropy was used to capture the inheritance information. ANN was utilized to generate the forecast, and the proposed model was found to be resilient and effective in predicting ISMR.

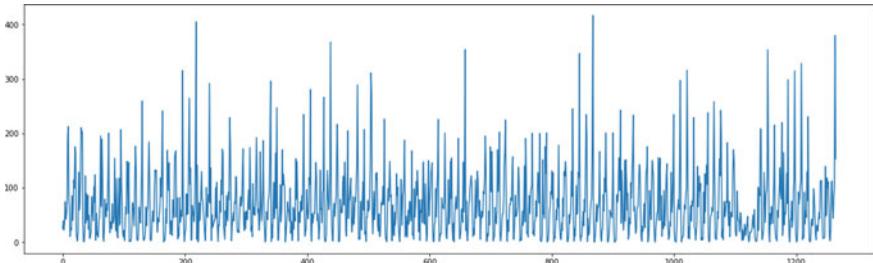
Dash et al. [6] employed three neural network models for predicting rainfall during monsoon. They used sea level pressure (SLP), sea surface temperature (SST), and by combining SST with SLP (SST + SLP) as input predictors. Their findings indicated that ELM outperformed all other methods in predicting Indian summer monsoon rainfall.

Elsworth and Güttel [7] suggested a novel approach by combining machine learning techniques and text generation for predicting time series. Their research demonstrated that by utilizing this combined approach, the training phase can be significantly accelerated without sacrificing the accuracy of the predictions, provided that the time series are of adequate length. The study also showed that LSTM can be utilized for forecasting time series data.

Johny et al. [8] employed Adaptive Ensemble Empirical Mode Decomposition and Artificial Neural Network to forecast Indian summer monsoon rainfall. This hybrid model is designed to adaptively perform predictions as new information becomes available. The study compared the AEEMD-ANN model's performance with EEMD-ANN in predicting Kerala's monsoon rainfall.

Long Short-Term Memory (LSTM) employed by Dash et al. [9] was used to forecast Indian summer monsoon with sequential data. They provided normalized data as input to the deep LSTM network and predicted the value for the next time period by shifting one time period in the input series. The study found that the LSTM network has the ability to learn intrinsic weather data.

Compared with the above methods, deep learning [10, 11] is capable of accepting raw data and predicting the patterns in the dataset. This paper compares the performance of a deep learning sequential model, LSTM, with RNN in predicting monthly rainfall using time series dataset.



**Fig. 1** Time series plot for all India rainfall data from 1871 to 2016

### 3 Materials and Methods

This section provides description about the dataset and the methodology used.

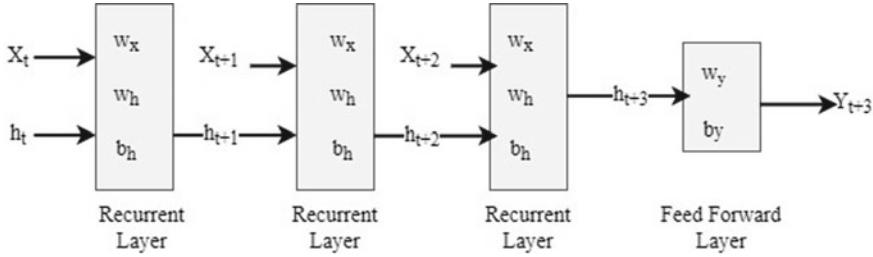
#### 3.1 Data Collection

To accomplish the aforementioned objectives, this study uses India's monthly rainfall data to develop a model. The dataset for all India monthly rainfall from 1871 to 2016 is collected from the Indian Institute of Tropical Meteorology, Pune [12] and is depicted in Fig. 1. This dataset was initially prepared and updated by various researchers [13, 14].

#### 3.2 Recurrent Neural Network (RNN)

RNN is utilized to analyze sequential data. While conventional feed-forward neural networks are suitable for data points that are not influenced by other data points, and RNN is specifically designed to capture the dependencies among sequence data. This is achieved by introducing a memory component in the network which stores information from previous time steps to generate the next time step in the sequence. Figure 2 depicts the diagram of the RNN architecture.

RNN has the flexibility to utilize various activation functions including sigmoid, tanh, and ReLu functions. The network calculates the value of hidden units and the output after k time steps in the feed-forward pass of RNN. The weights in the network are shared among the units. Two weights  $w_x$  and  $w_h$  are associated with the recurrent layer, where  $w_x$  for the input unit and  $w_h$  for the hidden unit.



**Fig. 2** Architecture of RNN

### 3.3 Long Short-Term Memory (LSTM)

LSTM [15] is an RNN variant tailored for handling sequences of data points. What makes LSTM unique is its ability to maintain a feedback connection and process sequences with lags of unknown durations between important events. This makes it well-suited for forecasting time series data. LSTM was designed to address the challenges of gradient explosion and vanishing gradients that commonly occur during the training of traditional RNNs.

An LSTM unit includes cell, input gate, forget gate, and output gate. Information flow within the cell is controlled by these three gates, and the value at any given time interval is stored in the cell. Figure 3 represents the architecture of LSTM.

Forget gate decides what amount of previous states  $h_{t-1}$  is allowed to move from block, input gate determines the amount of new data added to the recent input  $\chi_t$  and the output gate is responsible for determining the information to be passed on to the next level, as specified by the Eqs. 1–6.

$$\tilde{f}_t = \sigma(\omega_f \cdot [h_{t-1}, \chi_t] + b_f) \quad (1)$$

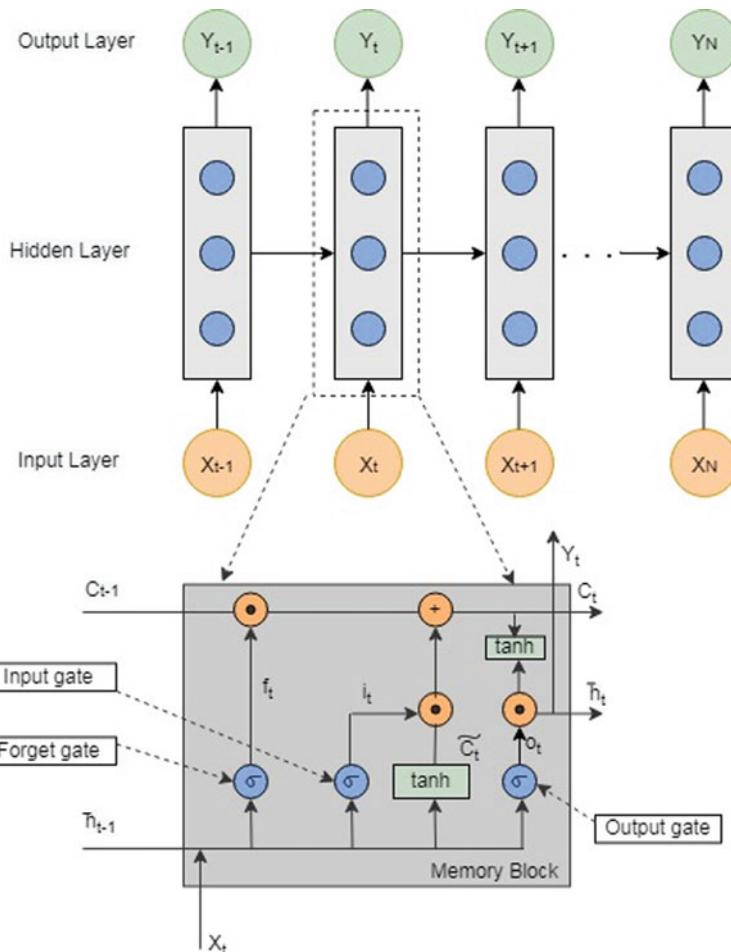
$$i_t = \sigma(\omega_i \cdot [h_{t-1}, \chi_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(\omega_c \cdot [h_{t-1}, \chi_t] + b_c) \quad (3)$$

$$\hat{C}_t = \hat{C}_{t-1} * \tilde{f}_t + \tilde{C}_t * i_t \quad (4)$$

$$o_t = \sigma(\omega_o \cdot [h_{t-1}, \chi_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(\hat{C}_t) \quad (6)$$



**Fig. 3** LSTM architecture

## 4 Experimental Results

The performance of the model in forecasting monthly rainfall is discussed.

### 4.1 Performance Evaluation Criteria

The efficiency is measured using the following criteria:  $o_i$  denotes the actual value,  $f_i$  represents the predicted value,  $\bar{o}$  corresponds to the mean of the original values,  $\bar{f}$  represents the mean of the predicted values.

**Mean Square Error (MSE)** quantifies the difference between the predicted and actual values as in Eq. 7.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (o_i - f_i)^2 \quad (7)$$

**Mean Absolute Error (MAE)** measures the overall difference among the predicted and actual values as in Eq. 8.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |o_i - f_i| \quad (8)$$

**Coefficient of Determination (R2)** suggests that the model's predictions are close to the actual values as in Eq. 9.

$$R^2 = \frac{\sum_{i=1}^N (o_i - \bar{o})(f_i - \bar{f})}{\sqrt{\sum_{i=1}^N (o_i - \bar{o})^2} \sqrt{\sum_{i=1}^N (f_i - \bar{f})^2}} \quad (9)$$

## 4.2 Results

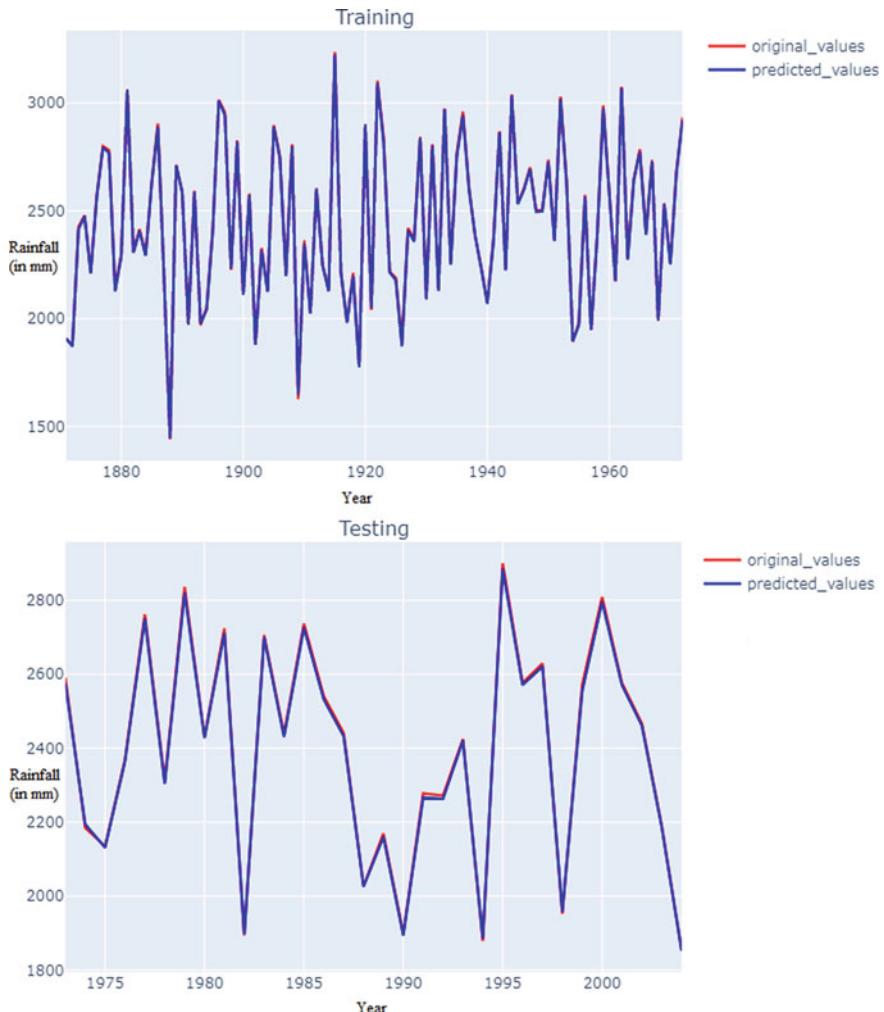
To create predictive models [16] for rainfall, the data from the past 12 years of rainfall values were utilized. An overlapping window of 12 years was employed to forecast the value for the next year ( $t + 1$ ). The input data for training the model was a 3D tensor of dimensions (1224, 12, 1), where the first parameter represents sample size, the next indicates the number of points in time, and the third represents the feature size. The remaining data was used as test set. 70% of the data was used to train the RNN and LSTM models, while the left over 30% was reserved for testing. The input layer had 11 neurons, the output layer had one neuron, and there were 2 hidden layers and each contain 50 neurons. Initially, the learning rate was set at 1.0.

Table 1 presents a comparison of the RNN and LSTM models based on the above-mentioned evaluation metrics. The findings indicate that the LSTM model outperforms the RNN model in terms of forecasting accuracy.

**Table 1** Performance of RNN and LSTM

Metrics	Existing model	Proposed model	
	AEEMD-ANN [8]	RNN	LSTM
$R^2$	0.91	0.941	0.954
MSE	0.96	0.496	0.466
MAE	84.28	69.89	53.23

The LSTM model exhibits superior performance over the RNN model, as indicated by its lower MSE and MAE values. Figure 4 depicts the time series plot of the forecasted outcomes.



**Fig. 4** Forecasting results of training and testing data

## 5 Conclusion

Accurate weather forecasting is crucial for planning daily activities, and deep learning models are utilized in numerous real-time applications, including weather prediction. This paper employs the RNN and LSTM models to forecast monthly rainfall across India using a time series dataset. The findings demonstrate that both models are effective in time series forecasting of monthly rainfall. Nonetheless, LSTM outperforms the RNN model in terms of forecasting accuracy, as evidenced by its lower MSE and MAE values and higher  $R^2$  value.

## References

1. Chattpadhyay S, Chattpadhyay M (2007) A soft computing technique in rainfall forecasting. arXiv preprint nlin/0703042
2. Chattpadhyay S, Chattpadhyay G (2008) Identification of the best hidden layer size for three-layered neural net in predicting monsoon rainfall in India. *J Hydroinf* 10(2):181–188
3. Kashid SS, Maity R (2012) Prediction of monthly rainfall on homogeneous monsoon regions of India based on large scale circulation patterns using genetic programming. *J Hydrol* 454:26–41
4. Bianchi FM, Maiorino E, Kampffmeyer MC, Rizzi A, Jenssen R (2017) An overview and comparative analysis of recurrent neural networks for short term load forecasting. arXiv preprint [arXiv:1705.04378](https://arxiv.org/abs/1705.04378)
5. Singh P (2018) Indian summer monsoon rainfall (ISMR) forecasting using time series data: a fuzzy-entropy-neuro based expert system. *Geosci Front* 9(4):1243–1257
6. Dash Y, Mishra SK, Panigrahi BK (2019) Neural network based approaches for prediction of the Indian summer monsoon rainfall. In: 2019 2nd International conference on intelligent computing, instrumentation and control technologies (ICICICT), vol 1. IEEE, pp 550–554
7. Elsworth S, Güttel S (2020) Time series forecasting using LSTM networks: a symbolic approach. arXiv preprint [arXiv:2003.05672](https://arxiv.org/abs/2003.05672)
8. Johny K, Pai ML, Adarsh S (2020) Adaptive EEMD-ANN hybrid model for Indian summer monsoon rainfall forecasting. *Theoret Appl Climatol* 141(1):1–17
9. Dash Y, Mishra SK, Panigrahi BK (2019) Indian summer monsoon prediction by deep long short-term memory neural network using time series based approach. In: AGU fall meeting abstracts, vol 2019, pp A33M-2968
10. Aswin S, Geetha P, Vinayakumar R (2018) Deep learning models for the prediction of rainfall. In: 2018 International conference on communication and signal processing (ICCSP). IEEE, pp 0657–0661
11. Yen MH, Liu DW, Hsin YC, Lin CE, Chen CC (2019) Application of the deep learning for the prediction of rainfall in Southern Taiwan. *Sci Rep* 9(1):1–9
12. <http://www.tropmet.res.in>. Accessed 22 Jan 2023
13. Parthasarathy B, Munot AA, Kothawale DR (1994) All-India monthly and seasonal rainfall series: 1871–1993. *Theoret Appl Climatol* 49(4):217–224
14. Kothawale DR, Rajeevan M (2017) Monthly, seasonal, annual rainfall time series for All-India, homogeneous regions, meteorological subdivisions: 1871–2016
15. Sherstinsky A (2020) Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D* 404:132306
16. Kala A, Vaidyanathan SG (2022) Forecasting monthly rainfall using bio-inspired artificial algae deep learning network. *Fluct Noise Lett* 21(02):2250018

# Detection and Classification of Dental Caries Using Deep and Transfer Learning



Divya Rajput, Hiral Rane, Devika Nikam, Janhavi Wagh, and Anuja Jadhav

**Abstract** Dental caries is an immensely frequent complication in dental sector which creates a greater impact on the majority portion of the population. Dental caries is very herculean to detect as their location makes clinical analysis strenuous. Compared to traditional methods, the modern approaches are faster, less labor-intensive, and more precise. Advanced computation algorithms and transfer learning models are increasingly utilized in dentistry to improve the efficiency of detecting and classifying periapical lesions and caries. The data augmentation and automatic feature extraction process for training and testing classification models include multiple iterations. Nonetheless, inaccurate results may hinder the diagnostic procedures. To help dentists enhance the efficiency of caries detection, cutting-edge computing techniques along with pre-trained architecture like VGG-16 are utilized. In this paper, the optimal outcomes were accomplished by VGG-16 algorithm with steady learning rate of 0.001 and Adam optimizer displaying accuracy on the validation data as 98.70. The execution provides a wider objective and application of the model in dentistry.

**Keywords** Artificial intelligence · Deep learning · Transfer learning (TF) · Pre-trained models · Convolutional neural network (CNN)

---

D. Rajput (✉) · H. Rane · D. Nikam · J. Wagh · A. Jadhav

Department of Information Technology, Pimpri Chinchwad College of Engineering, Pune 411044, India

e-mail: [divya.rajput19@pccoepune.org](mailto:divya.rajput19@pccoepune.org)

H. Rane

e-mail: [hiral.rane19@pccoepune.org](mailto:hiral.rane19@pccoepune.org)

D. Nikam

e-mail: [devika.nikam19@pccoepune.org](mailto:devika.nikam19@pccoepune.org)

J. Wagh

e-mail: [janhavi.wagh19@pccoepune.org](mailto:janhavi.wagh19@pccoepune.org)

A. Jadhav

e-mail: [anuja.jadhav@pccoepune.org](mailto:anuja.jadhav@pccoepune.org)

## 1 Introduction

Healthcare sector is a consequential fragment of the nation. The maximum part of the population suffers from oral dental disease problem. Dental care these days is expensive, and majority of people neglect its consequences [1]. The caries identification is an eminent research complication in medical dentistry sector; it depends on computing power and pervasive technology to detect dental problems that if not treated could further lead to periapical lesions. It aims at accurately identifying the area of teeth so that specific areas can be targeted and addressed. Currently, the rapid and ever-increasing population and scattered medical resources have caused precision medical care in dentistry industry to capture remarkable attention of researchers [2]. The deteriorating dental conditions lead to oral diseases like cavities, lesions, oral cancers, and dental injuries. So, caries identification is necessary, and it is significant to come up with techniques to fathom dental diseases endured by distinct individuals.

In spite of the challenges discussed in the above problem statement, caries disease detection and identification are yet a functioning and active area of research. Numerous techniques have been developed over the years. The traditional classification algorithm like support vector machine [3] relies heavily on preprocessing and is implemented in dental detection depending on the type and stage of disease to predict the classification accuracy. The consumption of sugary diet which comprises foods and drinks, alcohol, and tobacco is yet another reason for the enhancement of dental diseases [4]. Thus, it is noteworthy to initiate an effective deep learning approach that will provide an effective solution and methodology. CNN-based method is thus exploited for tooth illness diagnostic in medical imaging context. One of the drawbacks in applying CNN-based approaches is applying such models requires abundant amounts of datasets to train the machine. Yet another constraint is that a large number of image acquisition and processing requires a lot of processing power (GPUs) and is time-consuming [5]. Even factors like distinct plant backgrounds while capturing images, changes in light conditions, and other external environmental problems can compromise the results and evaluation measures. As technological growth is accelerated and with availability of absolute infrastructure, we can diagnose this problem and increase caries detection productivity and sustainability. On victorious accomplishment of the expected output the model will deliver the functionality that will enable it to detect the dental disease in real-time.

## 2 Background

This section presents the VGG-16 convolutional neural network (CNN) which is utilized to design the proposed model.

## 2.1 VGG-16 Convolutional Neural Network Algorithm

Convolutional neural network is an algorithm which uses deep learning to extracts input visual images and avail convolutional operations rather than matrix multiplication [6]. The VGG-16 is a transfer learning model that comprises ImageNet dataset, and its infrastructure is based on convolution neural network, with distinct structure organization. The network is abundant and encompasses approximately 138 million parameters [7] as shown in Fig. 1.

The above image depicts VGG-16 architecture, and the input image has the output shape of tensor (224, 224, 3). The initial two convolution layers of block1 have output shape (224, 224, 64); therefore, the filter size is  $3 \times 3$  with stride value 2 and similar padding (size of image will remain constant). The pooling layer of block1 is allocated  $3 \times 3$  filter size and strides over a matrix in 2 steps. The block2 has 2 convolution layers with output shape (112, 112, 128) and max-pooling (56, 56, 128) with 128 filters. The 3 convolution layers of block3, block4, and block5 followed by max-pooling layers stack have 256, 512, and 512 filters, respectively, each with size of (2, 2) with same padding. The obtained feature map is flattened into feature vector. The hidden layers are assigned ReLU as an activation function, while output layer is assigned softmax [8].

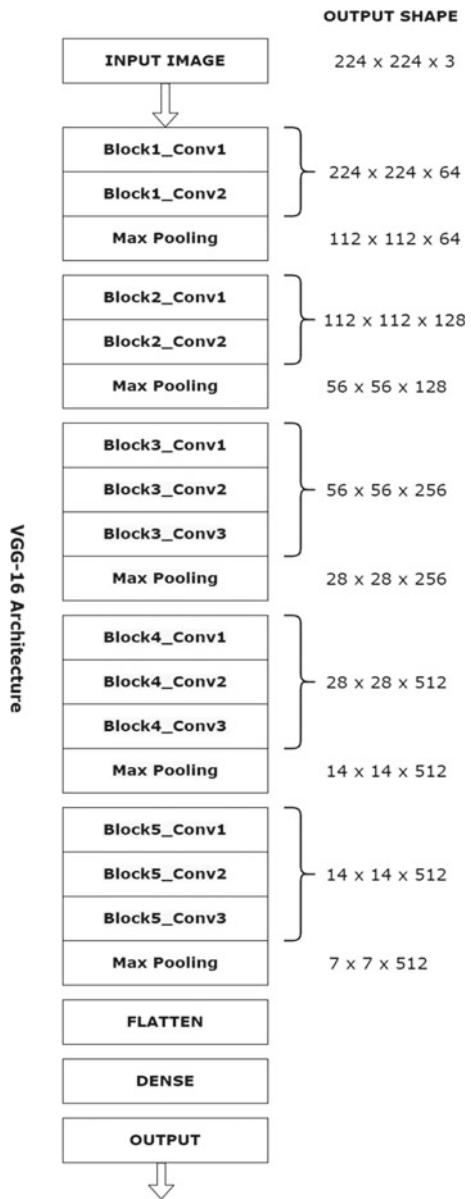
## 3 Literature Survey

In this segment, perspectives of different types for identifying caries using various disease detection approaches are analyzed.

Moran et al. [9] concentrate on conferring a novel approach that combines a strategy called picture processing strategies and CNNs to become aware of closely attached periapical lesions in radiographic snapshots and segregate them based on lesion extremity along with the aid of acquiring 112 bitewing radiographs. They extracted individual teeth photographs, and from examining these images, they further carried out an image data generation and used the gained pictures to train convolution neural networks. They evaluated the image dataset on the pre-trained Inception and ResNet architectures, and the implementation of these algorithms followed by several iterations during the training process and testing phase helped gain knowledge and understanding of the models, and varying optimal results were obtained and evaluated.

Mitra and Tarnach [10] represented that artificial intelligence has an extensive range of applications in remedy and dentistry, from fact processing and statistics retrieval to the usage of neural networks for diagnosis and the incorporation of augmented fact and virtual fact into dental training. Artificial intelligence (AI) is being studied in dentistry for a spread of functions, including the identity of regular and anomalous structures, sickness diagnosis, and remedy outcome prediction. This review looks at some modern-day and future programs of AI in dentistry. AI is undeniably the future of dental coaching control.

**Fig. 1** Architecture of VGG-16



Schwendicke et al. [11] supervised the general concepts of deep learning algorithms and the way to enhance them. The caries lesions are labeled and noted in a pixel-wise manner. The CNN U-Net is the methodology executed. The overall analysis is performed using validation metrics. The medical image is categorized into

training set, validation set, and test dataset. Performance of the CNN model exhibited powerful sensitivities for initial and advanced lesions. When the comparative analysis was performed, the conclusion is derived where that the radiographs were more effective than dentists.

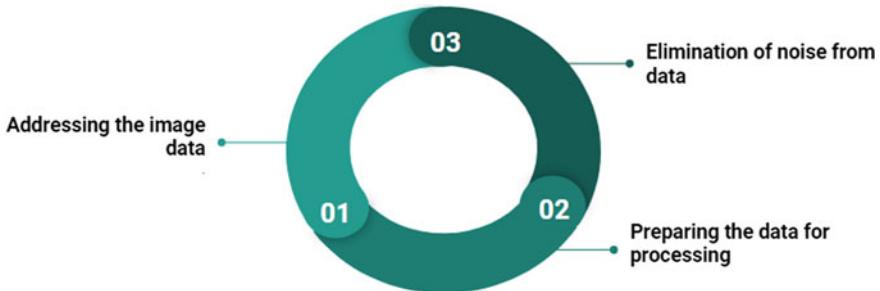
Li et al. [12] focused on appointing one of the fastest-growing laptop technologies available today to assist streamline the choice-making procedure and offer a professional-backed recommendation to the treating practitioner. Synthetic intelligence (AI) has been integrated into lots of components of each day's lifestyles and proved superior to the human mind in regard to selection-making competencies. To develop a neural network algorithm, 182 completed orthodontic cases of sophistication III malocclusion sufferers beyond their growth spurt have been selected from the records of the College of Illinois, Chicago, and sufferers' assembly of the inclusion standards and having good enough pre- and publish-treatment facts were further analyzed at their pre-orthodontics level.

Endres et al. [13] focuses on the most repeated results in dentistry the periapical radiolucencies which are identified with the use of panoramic radiographs. The comparison of the findings is displayed using a deep learning algorithm for prediction using radiographic datasets. The dataset in the medical domain is limited thus maintaining decent true positive rate, and attaining better performance along with better F1 score and average precision is difficult task. All the monitoring and evaluation parameters mentioned above like F1 score, precision, and true positive rate are calculated to derive best algorithm. The research paper focuses on assisting the surgeons as well dentists in detecting periapical lesions and caries on the panoramic radiographs.

## 4 Motivation

In taking this topic, we are motivated by the fact that it is one of the most noteworthy and distinguished problems, which if addressed constructively can serve as a bridge to greater success in the medical and healthcare domain. Oral health care is a ubiquitous requirement; there exists a perpetual chance for each independent person to commit and ingrain numerous strategies to untether an ideal deep framework with maximal coherence [14]. The deep neural along with transfer learning network mimics the human brain, thus yields outstanding accuracy. Scientific vital procedures that assist experts in inspecting caries are needed due to high incidence of caries lesions. Timely pinpointing is pivotal factor for beginning effectual cure for dental diseases.

The goal of the paper is to help doctors by enhancing productivity, life cycle, and efficiency of the medical diagnosis so that it leads to overall oral health condition growth in the healthcare sector. Delay in treatment leads to frequent detection of caries lesions in further stages which is subjected to restorative treatment as the only optimal treatment. The diagnosis and classification of caries encompass both clinical examination and radiographic image dataset examination and interpretation. The AI-based technology has far-reaching dominance over extensive applications in



**Fig. 2** Working on data

the field of dental medicine. It exhibits exceptional solutions to common problems in dentistry like dental caries. The objectives of the paper are determination of difficulty level of the case and suggesting whether or not the case be referred to a specialist before initiating caries treatment, deploying efficient algorithms to carry out caries detection and classification, and performing overall analysis on the deployed models.

#### **4.1 Proposed Methodology**

The proposed paper all in all relies on coherent dental caries detection and classification. A range of orthodox algorithms is utilized to examine and analyze the conclusions obtained including KNN, DT, and RF. The latest algorithms constructed on CNN model, transfer learning, and pre-trained architecture are implemented to measure the differences. The following three steps are followed for data preparation as shown in Fig. 2.

##### **4.1.1 Working on Data**

These steps are repeated and evolved till the desired result is achieved. We can also tweak or change the steps depending on our requirements. We can even reconsider the arrangements [15].

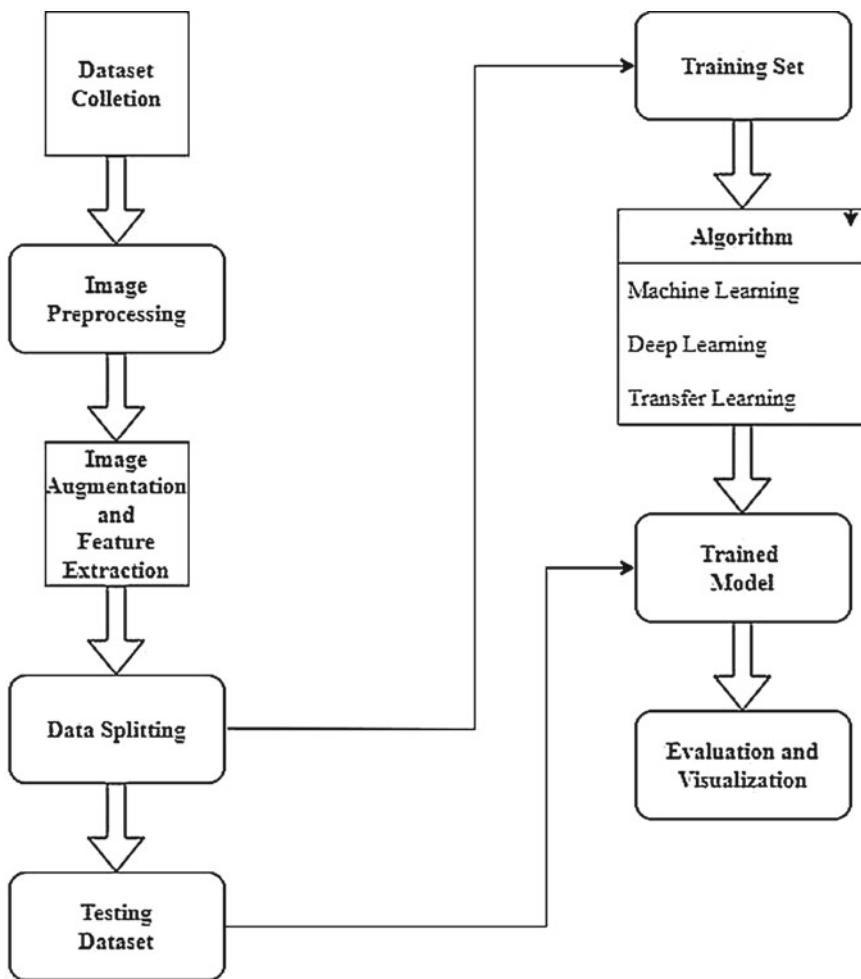
Data is the integral part of machine and deep learning algorithms, respectively. Obtaining and working with the data properly can result in significant improvements in the model. It includes following points like data collection, data inspection, and summary statistics (helps to recognize shape, scale and trend in data), and data cleaning is performed to eliminate the inconsistencies and filtering the noise.

Over the time, various diverse techniques and algorithms are executed to acknowledge dental disease identification and detection.

#### 4.1.2 Flowchart

The work flow of the proposed model is shown in Fig. 3 [16].

For this paper, we have used the open source and public Dental Caries Dataset provided by Kaggle Repo. This dataset consists of approximately 100 of dental caries images for implementation of the algorithm. Few sorted images from dataset are displayed in Fig. 4.



**Fig. 3** Flowchart for proposed system



**Fig. 4** Sample images from the dataset

#### 4.1.3 Image Preprocessing

Preprocessing of images is the required step for disease discernment in dental imaging; it aids to achieve optimal precision and specificity. The raw dataset of images obtained is impure and noisy due to environmental conditions and human errors as well, so there is a need for elimination of these aspects. Image preprocessing is defined as the steps taken to compose images before feeding them to training data models like resizing, cropping, orientation, rescaling, color, and rescaling.

#### 4.1.4 Image Augmentation and Feature Extraction

Augmentation of images is an effective methodology that generates additional arbitrary images from preceding image samples. Each random image copy produced is distinct from others based on augmentation techniques like shifting, flipping, and rotating. So, massive collections of homogeneous images are created by implementing limited lines of code. Augmentation in keras is done using keras ImageDataGenerator class. Augmentation techniques like rotation, brightness, zoom, flips, and standardization are performed using it. The ImageDataGenerator function ensures that the algorithm is fed with a novel variety of data in the form of image at each epoch. Low memory utilization is yet another advantage.

Feature extraction is the speck of dimensionality reduction technique; it fetches noise-free and processed image dataset and reduces it into favorable groups. The attributes and features well suited to the dental images are taken into consideration. For instance, if the color features are withdrawn from the image dataset subsequently followed by texture features extraction, thereafter the task of feature selection is

accomplished. It is practiced to sort out the most appropriate features from an image dataset.

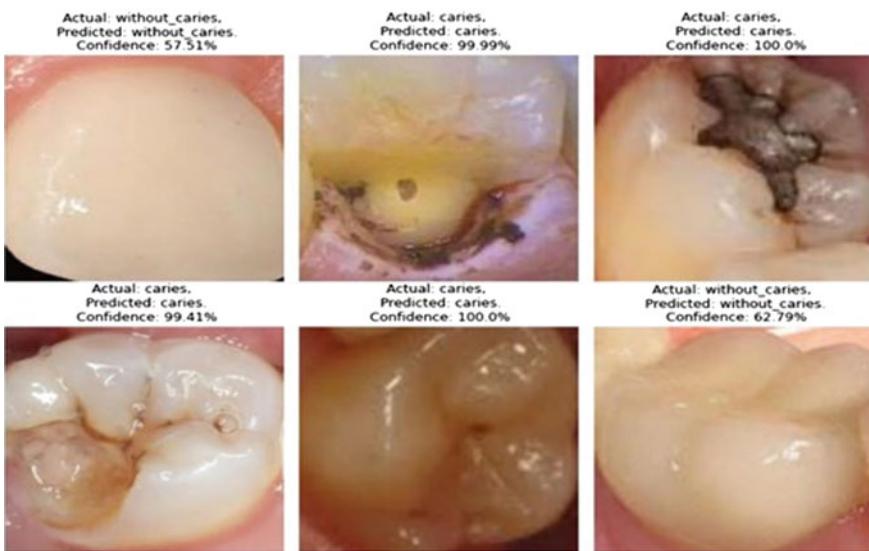
## 5 Results

The paper exploits divergent disease detection procedures utilizing deep neural networks and pre-trained models. This section converses the aggregates affiliated to the training segment of tooth data that accommodates implementation of working modules. The exhibited and shuffled image data is assessed to perform a statistical evaluation of actual versus predicted outcomes.

In Fig. 5, the images depict the actual image labeling and the obtained, i.e., the predicted label of the dataset after training. The confidence is defined as the score that explains how confident the fundamental algorithm deployed is and how assertive the correct value is drawn out. From the first six tested images obtained from the batch, the confidence is varying, the hyperparameters are updated, and iterations are performed to attain higher confidence.

Table 1 concludes the results as follows: the highest training accuracy and testing accuracy are 100% using logistic regression and 100% using CNN and transfer learning; the results obtained are due to the limited number of images in the dental dataset. The training loss and testing loss are 8% and 4%, respectively.

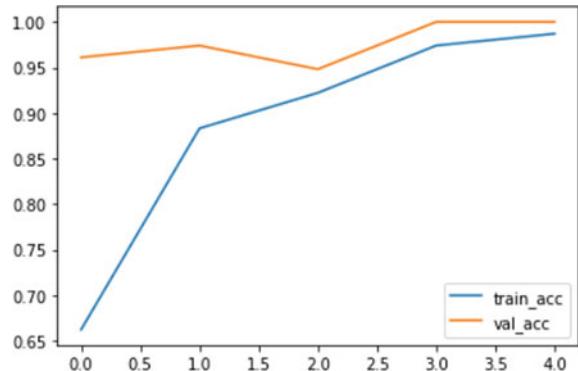
In Fig. 6, the iteration is conducted with epochs 5, and a pre-trained model is utilized to obtain the optimal result without exploiting many resources and computing



**Fig. 5** Implementation of working modules

**Table 1** Tabular comparison of all the implemented models

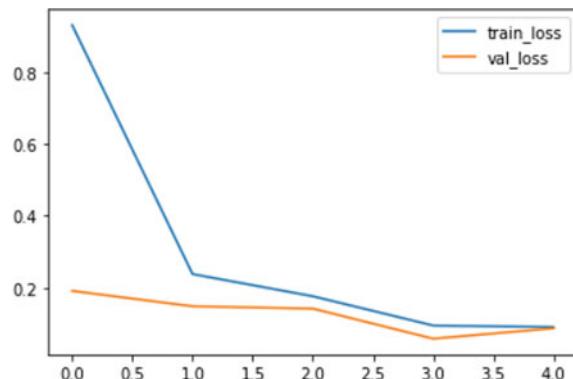
	Training accuracy	Training loss	Testing accuracy	Testing loss
SVM	93.22	7.8	80	20
Logistic regression	100	0.5	86.67	14.12
CNN (CPU)	86	23	91	16.64
CNN (GPU)	92	18.1	100	3.8
CNN (TPU)	96	15.2	100	6.8
Transfer learning	96.6	8	100	4

**Fig. 6** Training accuracy and validation accuracy of transfer learning algorithm

power. The testing accuracy is 100%, and the validation accuracy is 98.70%. To perform model compilation and optimization the Adam optimizer is used. The batch size is 16, i.e., 16 samples are used in a single epoch while training. The learning rate assigned is 0.001.

In Fig. 7, the training loss is 9% and the validation loss is 8.6%, respectively. The epoch's rate is 5. Batch size parameter is 16, and the learning rate is equivalent to 0.001 which is the significant hyperparameter in determining the efficiency of the model. The prior learnt on the similar model is re-employed on the secondary associated model. These approaches circumvent the dependency on huge data and provide enhanced effectiveness significantly reducing the training and validation loss.

**Fig. 7** Training and validation loss of transfer learning algorithm



## 6 Conclusion

To summarize, we can explicitly recognize that the deep learning and transfer learning models have the potential to aid dentists and alter the oral health sector by handling flaws that have been denounced in orthodox oral care. We have evaluated a pre-trained model for an effective caries classification task. The consolidated approach and faster amalgamation of deep learning algorithms in the healthcare sector benefits specialists and boosts their ability to treat and make intelligent decisions. The future scope involves the deployment of an end-to-end website for oral health detection.

## References

- Pandey P, Nandkeoliar T, Tikku AP, Singh D, Singh MK (2021) Prevalence of dental caries in the Indian population: a systematic review and meta-analysis. *J Int Soc Prev Commun Dent* 11(3):256–265. PMID: 34268187; PMCID: PMC8257015. [https://doi.org/10.4103/jispcd.JIS\\_PCD\\_42\\_21](https://doi.org/10.4103/jispcd.JIS_PCD_42_21)
- Oral Health in America (2021) Advances and challenges: executive summary [Internet]. National Institute of Dental and Craniofacial Research (US), Bethesda (MD). Available from: <https://www.ncbi.nlm.nih.gov/books/NBK576536/>
- Farhadian M, Shokouhi P, Torkzaban P (2020) A decision support system based on support vector machine for diagnosis of periodontal disease. *BMC Res Notes* 13. <https://doi.org/10.1186/s13104-020-05180-5>
- Tungare S, Paranjpe AG (2023) Diet and nutrition to prevent dental problems. [Updated 2022 Sep 9]. In: StatPearls [Internet]. StatPearls Publishing, Treasure Island (FL). Available from: <https://www.ncbi.nlm.nih.gov/books/NBK534248/>
- Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9(4):611–629. Epub: 2018 Jun 22. PMID: 29934920; PMCID: PMC6108980. <https://doi.org/10.1007/s13244-018-0639-9>
- <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>
- Tamma S (2019). Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *Int J Sci Res Publ (IJSRP)* 9:9420. <https://doi.org/10.29322/IJSRP.9.10.2019.p9420>

8. <https://www.geeksforgeeks.org/vgg-16-cnn-model/?ref=lbp>
9. Moran M, Faria M, Giraldi G, Bastos L, Oliveira L, Conci A (2021) Classification of approximal caries in bitewing radiographs using convolutional neural networks. Sensors 21:5192. <https://doi.org/10.3390/s21155192>
10. Mitra R, Tarnach G (2022) Artificial intelligence—a boon for dentistry. Int Dent J Student's Res 10:37–42. <https://doi.org/10.18231/j.idjsr.2022.009>
11. Schwendicke F, Oro J, Cantu A, Meyer-Lueckel H, Chaurasia A, Krois J (2022) Artificial intelligence for caries detection: value of data and information. J Dent Res 101. 220345221113756. <https://doi.org/10.1177/00220345221113756>
12. Li P, Kong D, Tang T, Su D, Yang P, Wang H, Zhao Z, Liu Y (2019) Orthodontic treatment planning based on artificial neural networks. Sci Rep 9(1):2037. PMID: 30765756; PMCID: PMC6375961. <https://doi.org/10.1038/s41598-018-38439-w>
13. Endres MG, Hillen F, Salloumis M, Sedaghat AR, Niehues SM, Quatela O, Hanken H, Smeets R, Beck-Broichsitter B, Rendenbach C, Lakhani K, Heiland M, Gaudin RA (2020) Development of a deep learning algorithm for periapical disease detection in dental radiographs. Diagnostics 10(6):430. <https://doi.org/10.3390/diagnostics10060430>
14. Oral Health in America (2021) Advances and challenges [Internet]. National Institute of Dental and Craniofacial Research (US), Bethesda (MD). Section 1, Effect of Oral Health on the Community, Overall Well-Being, and the Economy. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK578297/>
15. <https://www.analyticsvidhya.com/blog/2021/07/step-by-step-guide-for-image-classification-on-custom-datasets/>
16. Chiesa M, Maioli G, Colombo G, Piacentini L (2020) GARS: Genetic Algorithm for the identification of a Robust Subset of features in high-dimensional datasets. BMC Bioinformatics 21. <https://doi.org/10.1186/s12859-020-3400-6>

# Identification of Diabetic Retinopathy Using Robust Segmentation Through Mask RCNN



Aryan and Suman Deb

**Abstract** Medical imaging has come out to be very challenging field, in the area of computer vision; in this study, it has been aimed to detect diabetic retinopathy with image identification and segmentation techniques. Diabetic retinopathy (DR) is an eye disease caused by diabetes that can proceed to cause blindness; therefore, early detection is very critical to prevent visual disturbances. Evaluation will be done on the basis of mean average precision (MAP), aiming to detect lesions with the pretrained models like Mask-RCNN instance segmentation (R-50/R-101/X-101). It will finally come to the conclusion after custom training and testing on all set of real-life retina images taken from the clinical dataset and get a properly segmented region with nearly accurate bounding box. The model which identifies the accurate lesions caused by diabetic retinopathy with the disease named as “exudates” and “microaneurysms” will be considered for the future references.

**Keywords** Mask-RCNN · Image detection · Image segmentation · Lesions · Exudates · Microaneurysms · Deep learning · Convolutional neural network (CNN)

## 1 Introduction

Diabetic retinopathy causes a major issue; eventually, it leads to pragmatic disaster if not cured and spotted at very early detection. It leads to different types of vision issues in the patient’s eye, commonly known as cotton wool spots, exudates, microaneurysms, haemorrhages and abnormal growth of blood vessels.

Modern-day doctors are facing the similar kind of issues considering it to be very time-consuming process to filter out the images which is not very eye catchy because of low visibility of tiny spots; therefore, it becomes very inefficient to detect

---

Aryan (✉) · S. Deb  
Department of CSE, NIT Agartala, Paschim Barjalai 799046, India  
e-mail: [aryan215gupta@gmail.com](mailto:aryan215gupta@gmail.com)

S. Deb  
e-mail: [sumandeb.cse@nita.ac.in](mailto:sumandeb.cse@nita.ac.in)

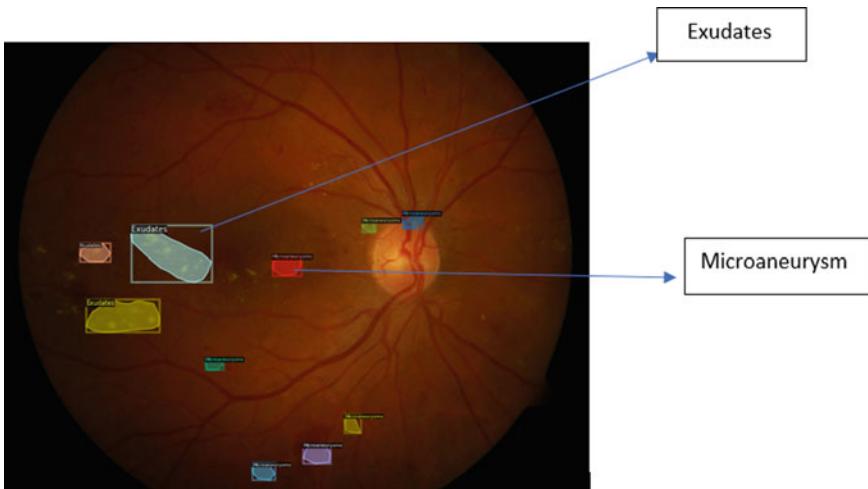
the lesions. To counter this inefficiency, a large number of Kaggle challenges are exposed globally to make process more practical.

With the help of machine learning, deep learning, computer vision which plays a very important and pivot role to counter these kind of issues, our aim is to identify and segment the images of diabetic retinopathy using Mask-RCNN models based on the average precision value.

## 1.1 Goals

Our main aim here is to identify and segment the image with the nearly accurate bounding box which detect's the lesions caused by diabetic retinopathy named as exudates and microaneurysm as shown in Fig. 1. We are motivated to identify diabetic retinopathy with different pretrained models of Mask-RCNN.

Apart from this, our aim is to experiment with different layers of Mask-RCNN, checking with different models named as R-50, R-101, X-101. Evaluating each parameters which will end up giving the final image. Finally, testing with some unknown dataset of retina image will decide which model acquires the best identification and segmentation.



**Fig. 1** Masking of exudates and microaneurysms

## 2 Related Work

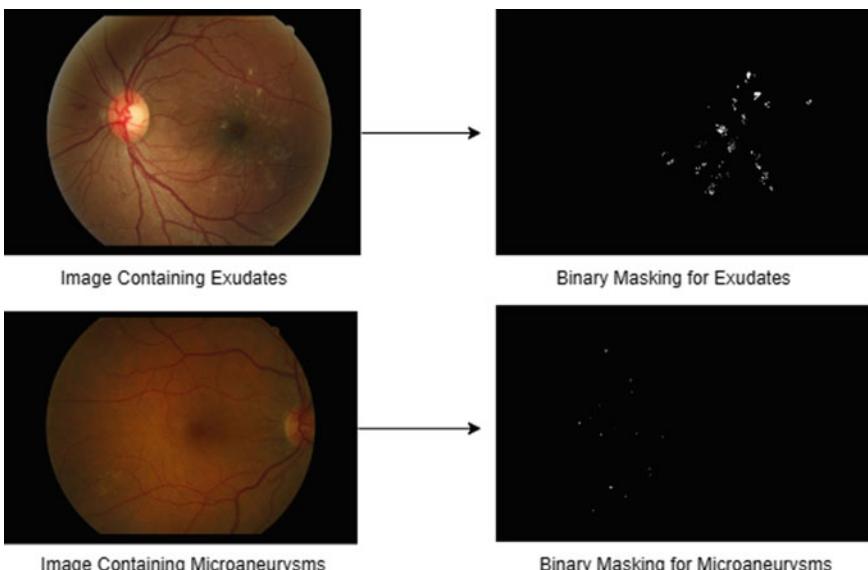
The binary mask segmentation over the image containing exudates and microaneurysms with the colour that has been normalized with the weighted pixels as shown in Fig. 2 is presented by shenavarmasouleh, Farzan and Arabnia [1]. Then, the author has applied Gaussian blur with sigma ( $X = 20$ ) on it and it has been added to the original version. The author is assigning the weights of 4 and -4 to the original and blurred images, respectively. The gamma was also set to 128. Finally, the images were resized to  $1024 \times 1024$  pixels.

Another work done in [2] shows that after the completion of pre-processing stage, the images have been extracted from the Kaggle dataset in order to dilute with the Mask-RCNN to generate some of the information (mask of each region/images, bounding box, classes and some of the other extra added features). It has been divided into two phases [3].

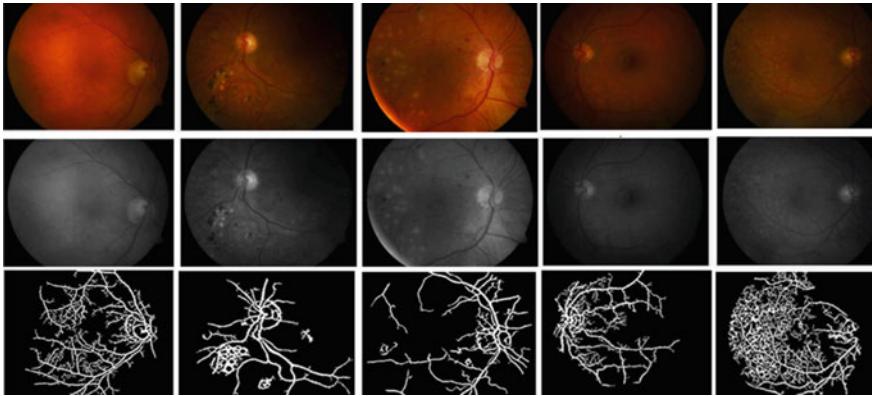
In phase-1, the aim is to use E-optha dataset to train and evaluate the performance of Mask-RCNN.

In phase-2, we are also taking the advantage of the pretrained model, which we discussed in phase one in order to detect the severe damages caused in retina.

Utilizing the Mask-RCNN [4] for image segmentation has been recently used to segment the oral disease [5]. The author has applied some deep learning algorithms which is able to segment the particular region which is causing the number of oral diseases that are in the form of Thrush, Leukoplakia, Lichen planus, etc. Masking



**Fig. 2** Mask-RCNN binary masking for exudates and microaneurysms



**Fig. 3** Unusual growth of blood vessels

the region of oral cavity gives the accurate bounding box region, and the extent the region of oral cavity has to be filled.

The author has taken the input retina image and presented the visual result of optic disc segmentation with the proper annotated image which result in detecting the glaucoma from the input retina image [6].

Another work has been done on the basis of medical imaging diagnosis using image segmentation on the unusual growth of blood vessels as shown in Fig. 3, where the sample test images are obtained from the hospital [7]. The resultant images that have been classified in the process of segmentation are used to identify the abnormal growth of the blood vessels represented by the virtue of Mask-RCNN models. In Fig. 3, the first image is input image, the second image is greyscale conversion, and the third row is the segmented output [8].

The Mask-RCNN has also been ensembled for nuclei segmentation [9], the detailed analysis of combination of modules and its challenges over image segmentation has been done, and the result indicates that using the ensemble model in nuclei segmentation will improve the quality.

The first image is the input to the Mask-RCNN model, and the second and third images respectively will show the masked region of the given input. The model accuracy will be decided on the basis of intersection over union (IoU) and the mean average precision value (mAP).

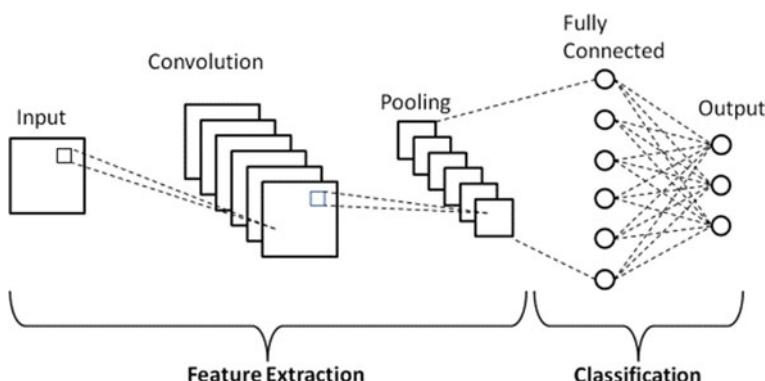
The another related work has been done over Mask-RCNN-based orange detection and segmentation which has been discussed in [10].

### 3 Methodology

Diabetic retinopathy causes major issue in human retina which itself is the sole problem of causing other deficiency in the patient's eye, for example exudates, haemorrhages, aneurysms, cotton wool spots and also immature growth of blood vessels causing many blind spots. Because it is now the major issue which is very common nowadays, the data has been publicly available which is mostly found online. The big and huge chunk of dataset often causes the learning complexities, for which, we need some valid amount of dataset with 62 images containing exudates and 179 images containing microaneurysms. E-optha dataset downloaded from "<https://www.adcis.net>" contains more than 300 images, in which we've only taken the limited image for our use case.

#### 3.1 Convolutions Neural Networks

An example of a deep learning neural network is a convolutions neural network (CNN) [11], which is frequently employed for image and video processing. The novel contribution of CNN architecture is shown in Fig. 4 where it is intended to process input through a number of layers [12], each of which is made up of a group of trainable filters. The edges, textures and shapes of the supplied data are extracted from it using these filters. For the network to learn spatial hierarchies of features from the input data, the filters must be involved with that data. Before moving on to the next layer, the output of each layer is often processed through a nonlinear activation function, such as ReLU. The network's final layers are often fully connected and used for tasks like categorization.



**Fig. 4** Convolution neural network

### 3.2 Understanding Mask-RCNN

Mask-RCNN [13] is a well-liked deep learning framework for computer vision tasks like segmentation. While Faster RCNN is used for bounding box object identification, fully convolutional networks (FCNs) are added to Faster RCNN to build masks for each item. A backbone, a region proposal network (RPN), a region of interest alignment layer (RoIAlign), a bounding box object detection head and a mask generation head can all be found in a Mask-RCNN.

A model called Mask-RCNN [14] was created for the segmentation of image instances. Instead of only extending Faster RCNN [15] to discover the bounding box and the class label, it goes a step further and focuses on creating pixel-level masks for each item. To the Fast RCNN model, it ingeniously adds a second FCN on top of RPN, resulting in a new parallel branch that generates a mask for the object located in a specified region. Notably, the authors had to somewhat alter the RoIPool to resolve the issue of location misalignment brought on by its quantization behaviour. The updated method was given the name RoIAlign. Further, it was given to the mask classifier which eventually gives the masked image of lesions (Fig. 5).

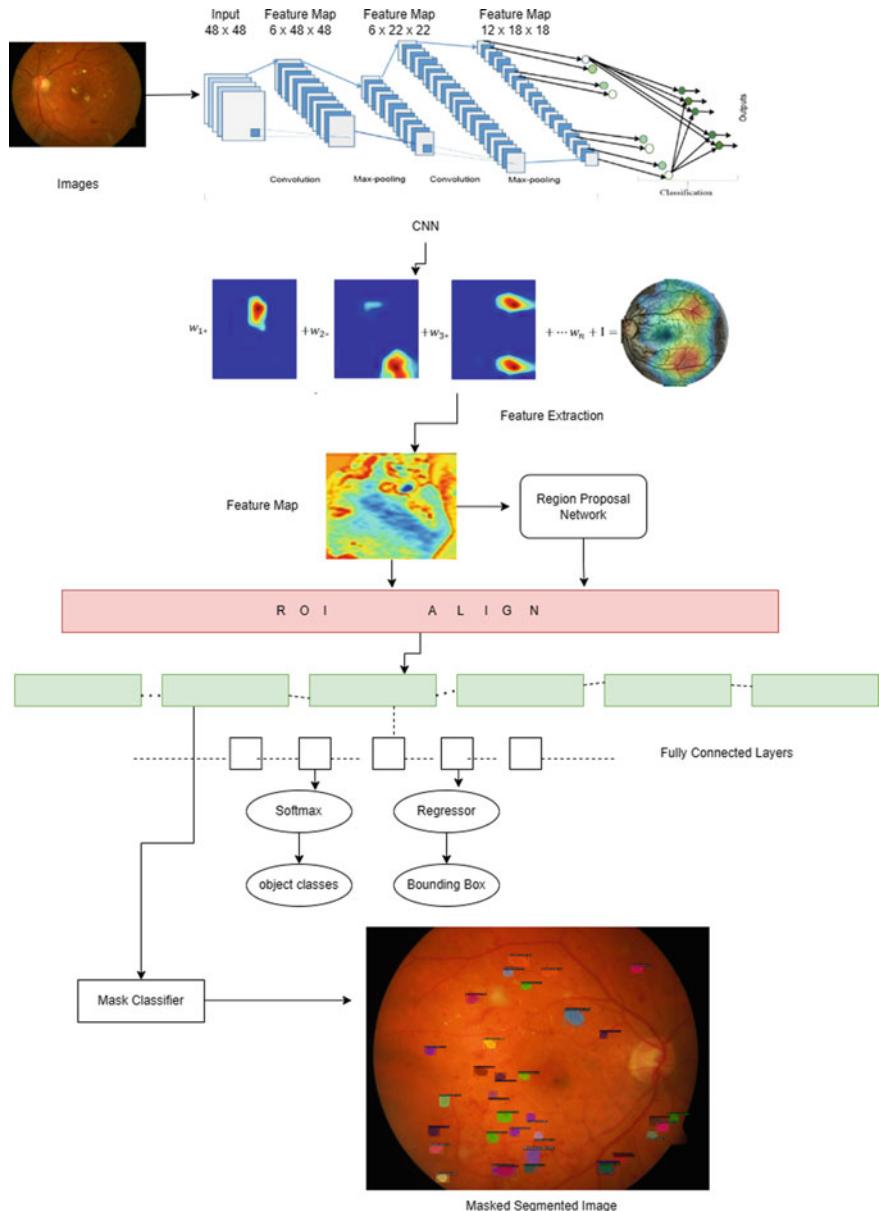
The model that has been used, each with the different architecture of the image segmentation will rightly give the result on the basis of custom-trained model, each with different number of epochs and iterations will lead to convergence of the input images with some annotated value, trained individually and masks the segmented region with exudates and microaneurysms, as shown in Fig. 6.

### 3.3 Model Comparison

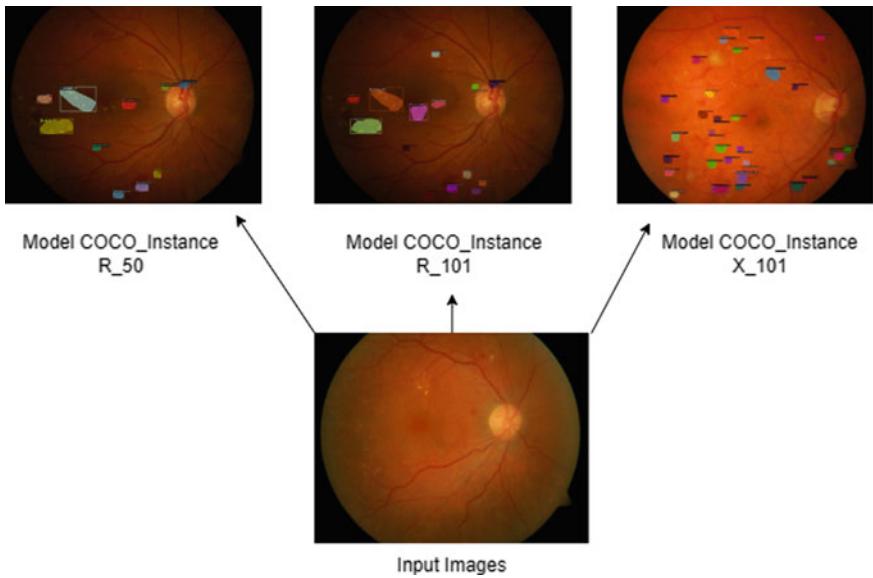
The comparison is based on the mask average precision (AP) versus inferences time, and other different parameters like learning rate, number of layers, number of iterations, no of epochs and so on as shown in Fig. 7. After checking with all the existing image segmentation model as captured in the figure below will tell us about the necessity to use the better accurate model like Mask-RCNN X-101 and further its sub-version compared with other model.

## 4 Evaluation and Testing

Before defining the (mAP), we should know about precision. Let us first define precision (P) which is able to calculate the percentage of correct predictions outcomes out of all predicted bounding boxes in the image. Average precision presents the performance of the models for each class while detecting the objects in a dataset. Finally, mean average precision will be taken as the average mean of the precision values across all the images in the dataset.

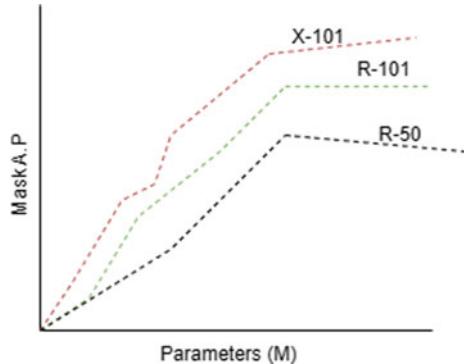


**Fig. 5** Architecture of Mask-RCNN module designed to detect lesions



**Fig. 6** Masking of exudates and microaneurysms

**Fig. 7** Comparisons of masking models



#### 4.1 Average Precision

It calculates the percentage of average precision of all outputs of the trained model. The average precision formula is defined in given Eq. 1.

$$\text{Average Precision} = \frac{1}{N} \sum_{i=1}^N \left( \frac{\text{TP}}{\text{TP} + \text{FP}} \right)_i \quad (1)$$

Here TP is true positive, FP is false positive,  $(\text{TP} + \text{FP})$  is the number of total true positive and false positive unit.

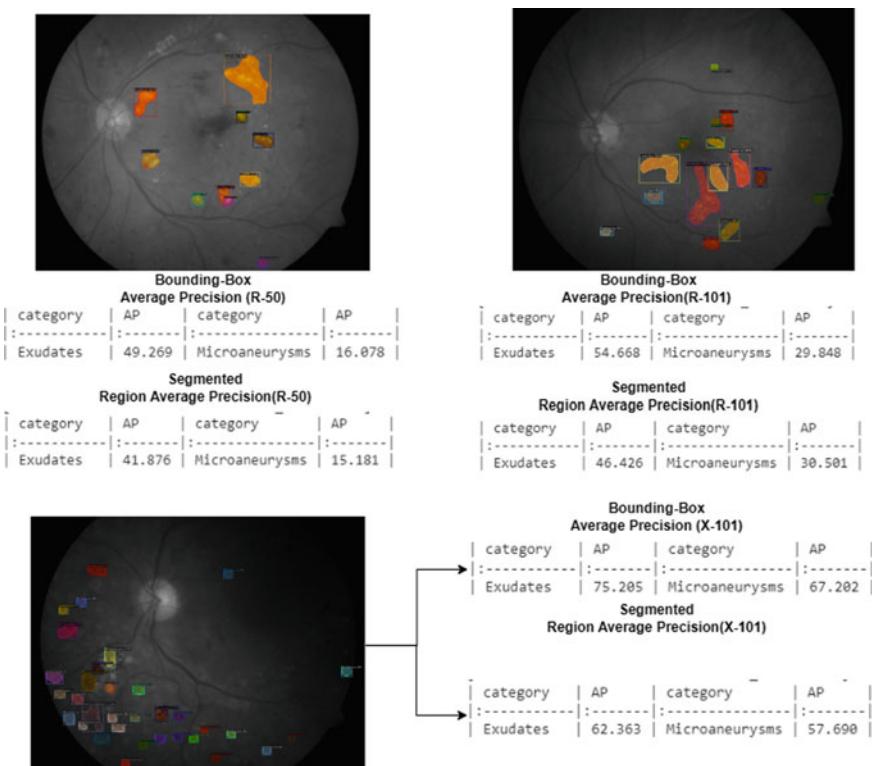
## 4.2 Testing

It has been randomly taken 4–5 images of retina for testing our Mask-RCNN model which has been trained over the large set of image. The efficiency of the model will depend on the accuracy of the testing images, whether it is properly segmented over the correct region of the lesions which will eventually decide the spot or the region of exudates and microaneurysms.

The evaluation will be tested on the basis of bounding box (BBox) [16] with it's mean average and precision (mAP ) and well as it's segmented region.

## 5 Conclusion

In this study, we have bundled several CNN-based image segmentation Mask-RCNN models and its respective architecture to determine the lesion caused in the retina named as exudates and microaneurysms, training it with Mask-RCNN models like



**Fig. 8** Detection of exudates and microaneurysms using Mask-RCNN with all the three custom-trained models, i.e. (R-50, R-101, X-101)

(R-50, R-101, X-101), and finally, it has been observed that the model X-101 gives the better bounding box average precision of “exudates” and “microaneurysms” which is around 75.20% and 67.20%, whereas the segmented region average precision value is around 62.363% and 57.690% which is effectively better than the model (R-50/R-101). Hence, it has been concluded that the model (X-101) performs better in both detection and segmentation, as shown in Fig. 8.

## References

1. Shenavarmasouleh F, Arabnia HR (2021) Drdr: automatic masking of exudates and microaneurysms caused by diabetic retinopathy using mask R-CNN and transfer learning. In: Advances in computer vision and computational biology. Springer, pp 307–318
2. Shenavarmasouleh F, Ghareh Mohammadi F, Hadi Amini M, Taha T, Rasheed K, Arabnia HR (2021) Drdrv3: complete lesion detection in fundus images using mask R-CNN, transfer learning, and LSTM. [arXiv:2108.08095](https://arxiv.org/abs/2108.08095)
3. Sopharak A, Dailley MN, Uyyanonvara B, Ann Barman S, Williamson TH, Thet New K, Aye Moe Y (2010) Machine learning approach to automatic exudate detection in retinal images from diabetic patients. *J Mod Opt* 57:124–135
4. Deb A, Chaudhuri R, Deb S (2023) An optimal approach for multi-class object detection. In: Distributed computing and intelligent technology: 19th international conference, ICDCIT 2023, Bhubaneswar, India, 18–22 Jan 2023, Proceedings. Springer, pp 335–340
5. Anantharaman R, Velazquez M, Lee Y (2018) Utilizing mask R-CNN for detection and segmentation of oral diseases. In: 2018 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp 2197–2204
6. Nazir T, Irtaza A, Starovoitov V (2021) Optic disc and optic cup segmentation for glaucoma detection from blur retinal images using improved mask-RCNN. *Int J Opt*
7. Das S, Kharbanda K, Suchetha M, Raman R, Dhas E (2021) Deep learning architecture based on segmented fundus image features for classification of diabetic retinopathy. *Biomed Signal Process Control* 68:102600
8. Gede Pande Darma I, Suardika IM, Maysanjaya D, Windu M, Kesiman A (2022) Optic disc segmentation based on mask R-CNN in retinal fundus images. In: 2022 4th International conference on biomedical engineering (IBIOMED) pp 71–74
9. Oskar Vuola A, Ullah Akram S, Kannala J (2019) Mask-RCNN and u-net ensembled for nuclei segmentation. In: 2019 IEEE 16th International symposium on biomedical imaging (ISBI 2019). IEEE, pp 208–212
10. Ganesh P, Volle K, Burks TF, Mehta SS (2019) Deep orange: mask R-CNN based orange detection and segmentation. *IFAC-PapersOnLine* 52(30):70–75
11. Albawi S, Abed Mohammed T, Al-Zawi S (2017) Understanding of a convolutional neural network. In: 2017 International conference on engineering and technology (ICET). IEEE, pp 1–6
12. O’Shea K, Nash R (2015) An introduction to convolutional neural networks. [arXiv:1511.08458](https://arxiv.org/abs/1511.08458)
13. Bharati P, Pramanik A (2020) Deep learning techniques—R-CNN to mask R-CNN: a survey. In: Das AK, Nayak J, Naik B, Pati SK, Pelusi D (eds) Computational intelligence in pattern recognition, Singapore. Springer Singapore, pp 657–668
14. Garifullin Azat, Lensu Lasse, Uusitalo Hannu (2021) Deep Bayesian baseline for segmenting diabetic retinopathy lesions: advances and challenges. *Comput Biol Med* 136:104725
15. Chen X, Gupta A (2017) An implementation of faster RCNN with study for region sampling. [arXiv:1702.02138](https://arxiv.org/abs/1702.02138)
16. Sultana Farhana, Sufian Abu, Dutta Paramartha (2020) Evolution of image segmentation using deep convolutional neural network: a survey. *Knowl-Based Syst* 201:106062

# Digital Watermarking Using Visual Cryptography



Alina Dash, Kshiramani Naik, and Sharmila Subudhi

**Abstract** Digital watermarking is a crucial information hiding or steganographic technique that can be infused with other security mechanisms, such as cryptography, to provide more robust security to data. The bit substitution method is primarily employed on a cover image to hide a secret watermark. However, they suffer from lossy compression, cropping, and other issues. In this work, we have proposed a two-out-of-two visual cryptography model that creates only two shares. These two visual secret shares cannot divulge any information on their own, but when used together, they reveal a secret image or watermark. Additionally, a random grid and an XOR operation are used for constructing these shares. A grayscale image is employed for embedding the watermark text in it. Later, the modified image undergoes the message extraction phase to reveal the watermark. Six transformations on the modified image have been done to highlight the behavior of the extracted watermark. A performance analysis is given for each such transformation. Moreover, a performance comparison of the developed model with another existing approach is also presented.

**Keywords** Steganography · Digital watermarking · Visual cryptography · Visual secret-sharing

---

A. Dash

Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, Odisha 768018, India

e-mail: [alinadash\\_cse@vssut.ac.in](mailto:alinadash_cse@vssut.ac.in)

K. Naik

Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Odisha 768018, India

S. Subudhi ()

Department of Computer Science, Maharaja Sriram Chandra Bhanja Deo University, Baripada, Odisha 768018, India

e-mail: [sharmilasubudhi@ieee.org](mailto:sharmilasubudhi@ieee.org)



(a) Visible Watermarking

(b) Invisible Watermarking

**Fig. 1** Digital watermarking

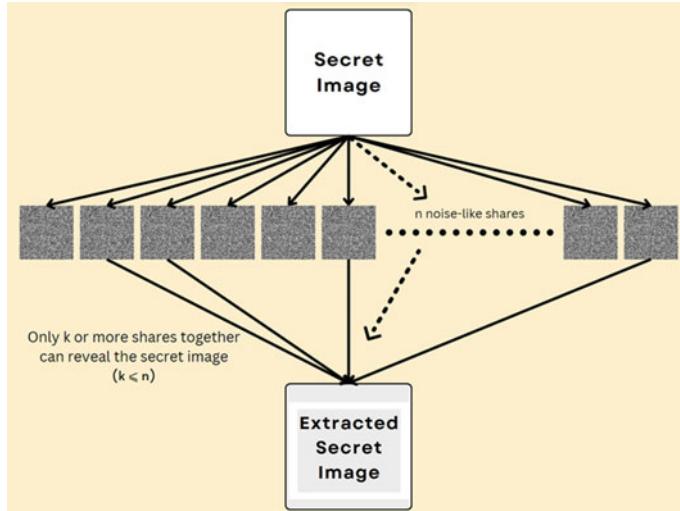
## 1 Introduction

Making information available to everyone worldwide has gotten far more accessible than it did in the past. That also implies that it would be simple for someone to utilize that information for personal purposes and then claim ownership of it. For instance, we may photograph a historic occasion digitally and consider selling it to the Times of India. Also, we may have distributed the same image to several other businesses to spur them into a bidding war. The image might then be slightly modified by one corporation employee, who would then claim ownership and effectively steal the image. Without digital data protection, the modern world will come to a halt. Digital watermarking comes to the rescue of the above and other such problems.

A digital watermark can subtly be incorporated into a signal, such as audio, video, or picture data that can tolerate noise [1]. It is generally done to establish an owner or have a copyright to a particular signal. Digital information is “watermarked” by being concealed inside a carrier signal. The embedded information may not be linked with the carrier. The validity of such a signal can be checked using watermarks and a way to reveal the owner’s identity [2].

Figure 1 presents two digital watermarking images. As we can see, the watermarking content through the naked eye is known as visible watermarking, as depicted in Fig. 1a. On the other hand, an invisible watermarking, as illustrated in Fig. 1b, is far more intriguing because the would-be thief is unaware that the material being stolen contains additional information. Invisible watermarking is unique because it can only be removed slowly [1].

Visual cryptography (VC) is used to achieve such a feat, where any encrypted message (image or text) that needs to be sent can be embedded inside multiple semi-transparent layers of a carrier image [2]. The encryption is done so that the original decrypted message can be received by sight reading. Furthermore, it is to be noted that visual cryptography is used to provide secure transmission in highly covert operations where the message is used only once, such as military and government businesses [2].



**Fig. 2**  $(k, n)$  Visual cryptography method

Naor and Shamir [3] presented the VC method to secure a private picture among " $n$ " members. A binary secret picture is shared into " $n$ " noisy shadow images in a  $(k, n)$  threshold VC system (shown in Fig. 2). A transparency with a printout of each shadow image is given to the appropriate participant. Any " $k$ " out of " $n$ " participants can work together to expose the hidden picture throughout the revealing phase by stacking their shadows. The " $n$ " different images generated are known as shadows of secret images or shares, as presented in Fig. 2.

Research says that the human visual system can perceive a concealed image [4]. However, even with plenty of computer resources, nothing about the hidden image can be discovered with " $k - 1$ " or fewer shadows. A hidden picture is spread into " $n$ " shadows in a VC model [3]. The leftover " $n - k$ " shadows are non-necessities. To expose the hidden image, we require at least " $k$ " shadows, containing the key to revealing the secret. However, due to pixel inflation of shares, traditional VC algorithm makes the reconstructed picture larger than the original hidden image [2]. This results in a decreased visual quality of the exposed hidden image and enormous shadow sizes. Furthermore, this process uses a substantial amount of hardware resources leading to rising costs.

Therefore, a two-out-of-two VCS is suggested in the current approach to ensure less computational cost. Here, the two shares generated can give no information on their own but must be used together to discover the hidden image. Additionally, a random grid is used to generate a master share, while an exclusive-OR (XOR) operation is done on the pixels of the master share and the base image to build an owner share.

The article is laid out in the following manner. Section 2 throws light on the previous research in this familiar field. Section 3 pays attention to the proposed model and its working. A comparative performance analysis of our model is discussed in Sect. 4, whereas a summary of the contribution made is presented in Sect. 5.

## 2 Literature Survey

Thawkar [1] provided a method for copyrighted works and temper detection using an invisible picture watermark. The least significant bit (LSB) approach for inserting the watermark uses a secret key encryption algorithm. The verification (the extraction of the watermark) procedure uses the same encryption key, making it possible to copyright-protect digital assets, including photos, audio, and video. This model can recognize any changes made to the image's pixel structure.

Shamir and Naor [3] developed a “k-out-of-n” ( $k < n$ ) visual secret-sharing (VSS) model that provided base for digital watermarking approaches. Here, one hidden image is dispersed into “ $n$ ” shares, out of which only “ $k$ ” or more shares can be kept together to discover it. The pixels of the black-and-white secret message are managed individually. So, each original pixel appears in “ $n$ ” modified versions as a collection of “ $m$ ” sub-pixels in close proximity. This makes human visual verification easier in a later stage because our eyes can average the black or white contribution and see the secret image.

Chandramathi et al. [4] provided encryption of visual data (images or text) in such a way that the human ophthalmic senses can handle their decryption without using any complicated cryptographic algorithms. This model divides a hidden image into shares that may be stacked until the secret image is revealed. Transparencies are typically used to display the shares. Yang [5] pointed out that using VSS is not expandable when the size of the original image is same as the shadow size.

Hsu and Tu [6] suggested a unique visual cryptographic digital watermarking system, where the watermark, a binary picture, is divided into “ $n$ ” shares using the VSS technique. After that, one of the shares is included in the host image while the owner holds the other. The owner must remove the embedded share and replace the watermark with his or her share to demonstrate ownership. These two shares cannot leak any information about the watermark due to the security feature of visual cryptography.

Abusitta [7] presented a watermarking model that chose random pixel values from an existing digital image instead of choosing specified pixels. Here, the watermark pattern need not be integrated with the original digital image. Instead, verification data, employed to confirm the ownership of the photograph, is created. As a result, the marked picture is now equivalent to the original image. The technique is based on the connections between randomly chosen pixels and their pixels' eight neighbors. Even if the attacker has modified the essential bits of a randomly chosen pixel, this connection maintains the marked image's coherence against various attacks. This

approach may recover the watermark pattern from the marked picture even when the original digital image has undergone significant alterations.

Das and Kundu [8] stated that the secrecy and authenticity of a message gets changed due to the assaults through frequent copying and alteration of the digital media material. Astuti et al. [9] presented a model that performed XOR operation three times to encode the hidden information before embedding on the pixel LSB. Three most significant bits (MSBs) are employed as keys for the XOR operations. Fatahbeygi and Tab [10] developed an image watermarking system that initially divides one carrier image in several non-conjoining blocks. Later, the support vector machine (SVM) along with canny edge detection is used to identify the smooth and edgy blocks among them. They have tried digital watermarking using visual cryptography to improve the robustness of their model by obscuring the watermark without altering the original carrier image.

Yang et al. [11] proposed a progressive VCS (PVCS) based on blocks, known as  $(2, n)$  block-based PVCS (BPVCS). A hidden picture is separated into “ $n$ ” separate, non-overlapping image blocks in  $(2, n)$ -BPVCS. The picture blocks corresponding to any “ $t$ ” participants will also be retrieved when “ $t$ ” participants stack the shadow images, where  $(2 < t < n)$ . Unfortunately, the cheating issue affects the performance of  $(2, n)$ -BPVCS model.

Li et al. [12] suggested a  $(t, k, n)$  XOR-based VCS (XVCS) to convert the hidden image in “ $n$ ” shadows (with “ $t$ ” essential and “ $n - t$ ” non-essential shadows). The message discovering procedure uses an XOR operation on these shadows. Bal et al. [13] suggested a model based on matching bit pairs and symmetric key cryptography (SKC) to hide a message inside a carrier image. Abdulazeez et al. [14] developed a hybrid robust watermarking model using lifting wavelet transformation (LWT), singular value decomposition (SVD), and multi-objective artificial bee colony optimization techniques.

Ren and Zhang [15] presented a distributed biometric storage mechanism to mitigate the security issues of centralized storage of biometric images in a cloud platform. They focused on improving the image quality while recovering them from existing VCS by implementing a transfer learning technique. Devi et al. [16] proposed a hybrid transformation model using the redundant discrete wavelet transform (RDWT) and singular value decomposition (SVD) to ensure the copyright protection of social media images. To protect such images and provide high imperceptibility and robustness, they have also employed the JAYA-Firefly (Ja-Fi) optimization algorithm. They have applied a DWT saliency map, power, and cosine functions to produce pseudo-random keys from the cover image. Further, these keys were used to encrypt the watermark by the random pixel position swapping (RPPS) process. Table 1 presents a summary of the studied literature.

**Table 1** Study of existing literature

Paper	Mechanism followed	Findings
Astuti et al. [9]	XOR operation is conducted three times to encode the message before embedding on the pixel LSB. Three most significant bits (MSBs) are employed as keys for the XOR operations	PSNR <sup>a</sup> value more than 50 dB and MSE <sup>b</sup> value not more than 0.3
Fatahbeygi and Tab [10]	Initially, the carrier image is divided into multiple non-conjoining blocks. The support vector machine (SVM) along with canny edge detection is used to identify the smooth and edgy blocks among them	PSNR value more than 35 dB
Yang et al. [11]	The $(2, n)$ BPVCS model separates a hidden picture into $n$ separated non-overlapping image blocks	Cheating issue affects its performance
Li et al. [12]	The $(t, k, n)$ XVCS converts the hidden image in $n$ shadows (with $t$ essential and $(n - t)$ non-essential shadows). The message discovering procedure uses an XOR operation on these shadows	Pixel inflation degraded the visual quality of the reconstructed image
Bal et al. [13]	Model used matching bit pairs and symmetric key cryptography (SKC) to hide the message	–
Abdulazeez et al. [14]	Developed a hybrid robust watermarking model using lifting wavelet transform (LWT), singular value decomposition (SVD), and multi-objective artificial bee colony optimization techniques	The quality of extracted watermark decreased with less number of decomposition levels
Ren and Zhang [15]	Presented a distributed biometric storage scheme with a cloud platform to resolve the security challenges faced by a traditional biometric storage unit	–
Devi et al. [16]	Proposed a hybrid transformation model using redundant discrete wavelet transform (RDWT) and SVD along with JAYA-Firefly (Ja-Fi) optimization algorithm.	PSNR value is around 46 dB

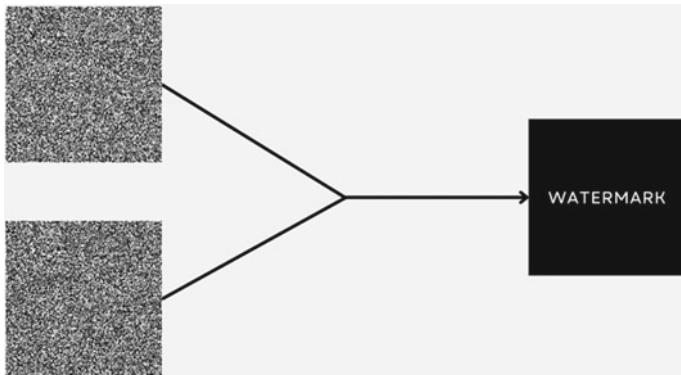
<sup>a</sup>PSNR = Peak Signal-to-Noise Ratio, represented in decibels (dB)

<sup>b</sup>MSE = Mean Square Error

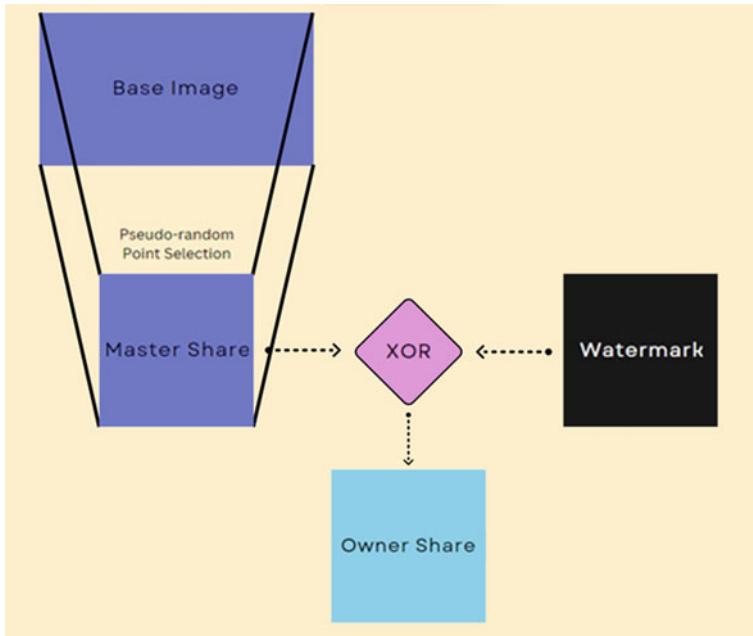
### 3 Proposed Method

Naor and Shamir [3] created the shares using the  $(k, n)$  visual secret-sharing (VSS) method to represent each pixel of the watermark as a matrix of “ $m$ ” sub-pixels, thus making all the generated shadows “ $m$ ” times the size of the watermark. This unnecessarily increased the computation cost. Therefore, to reduce the hardware requirements and eliminate the need for a third party, we propose a two-out-of-two  $(2, 2)$  VSS scheme without pixel expansion, where  $k = n = 2$ . Here, initially, two shares are generated that appear as noise individually, but when put together, it reveals the watermark. Figure 3 presents the mechanism behind extracting the watermark from two generated shares.

The share generated from the base image is the master share fused with a watermark to produce an owner share. The master share is generated using a set of pseudo-random points, as presented in Fig. 4. The key provided to the random module of Python ensures that the same set of random points is always selected from the base image. The mean of the intensity of the neighborhood pixels is compared to a threshold that decides if the respective pixel in the master share is black/white. So, every time we generate the master share, it is identical to its previous iteration.



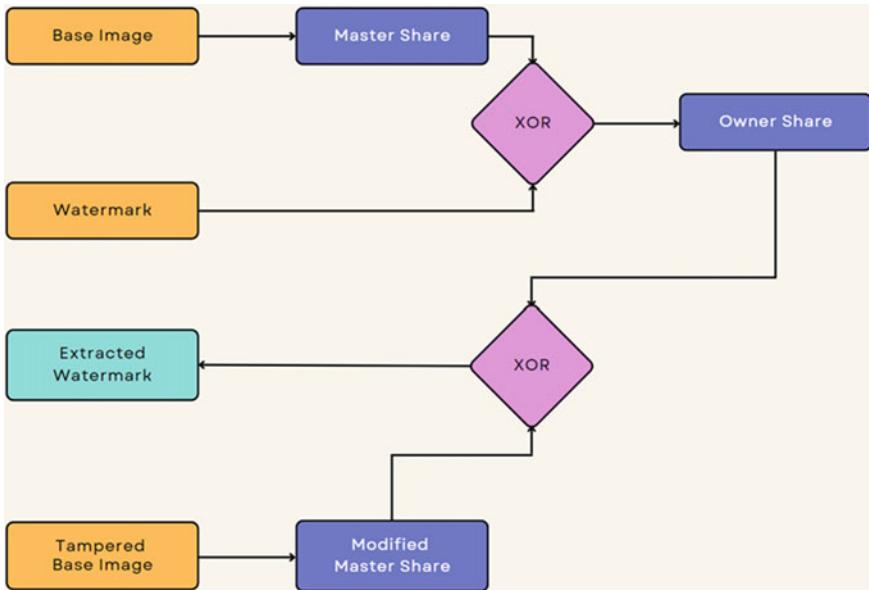
**Fig. 3** (2, 2) VSS model



**Fig. 4** Obtaining owner share using XOR method

Now, the corresponding pixels of the master share and watermark undergo an exclusive-OR (XOR) operation to generate the owner share. The owner share on its own is useless, but at any point, we can obtain the master share from the image and use both shares to verify authenticity, as seen in Fig. 4.

By making the " $m$ " subpixel the same size as the watermark pixel, i.e.,  $m = 1$ , the size of the shares is significantly reduced by " $m$ " times. This lack of pixel expansion allows us to perform the pixel-to-pixel operation using the XOR method.



**Fig. 5** Proposed watermarking system

The corresponding pixels of the master share and the watermark undergo XOR operation to build the owner share. This allows the owner share to appear as noise, but applying XOR to the pixels of the two generated shares gives us the watermark. This simple property of XOR makes both share generation and watermark extraction easier.

Moreover, the random point selection ensures minimal distortion to the generated master share and the extracted watermark. Therefore, when any attacker tries to make any changes to the image by adding a fake copyright symbol to claim it as their own, it will not be effective. Figure 5 depicts the proposed share generation and watermark extraction procedure.

A summary of the working steps is presented below. The suggested model is divided into two modules. The first module is responsible for generating the master and owner shares, and the second module is focused on watermark extraction.

### 3.1 Share Generation Module

1. Use a predefined key in the *random* module of Python to select a specific set of points in a base image.
2. Based on the mean neighbor intensity of a pixel and threshold value, paint the respective pixel in the master share as black or white.

3. The obtained master share pixels undergo XOR operation with the corresponding watermark image pixels to generate the owner share.

$$\text{owner\_share} = \text{master\_share} \oplus \text{watermark}$$

### **3.2 Watermark Extraction Module**

1. We can generate the master share from the base algorithm by choosing the same set of points.
2. The XOR operation is reversible, so we can obtain the watermark by performing the XOR operation of the corresponding pixels of the master and the owner share.

$$\text{owner\_share} \oplus \text{master\_share} = \text{watermark}$$

3. Even if the base image is modified, we can generate a master share and extract a distorted watermark by following step 2.

Being an application of invisible watermarks, any attacker can claim one image as its own by performing any transformations, such as a change in brightness or contrast, adding blur, or noise. The integrity of the extracted watermark under such transformations can be kept intact through our proposed system's random point selection mechanism. This ensures minimal distortion to the generated master share, thus making the watermark verifiable. However, the security of such images can get compromised, and extra information is needed to safeguard them.

## **4 Result Analysis**

We have taken an original image (Fig. 6a) and want to insert the watermark image (Fig. 6b) in it. The master share (Fig. 6c) is generated, and after having XOR operation, the owner share (Fig. 6d) is produced. Now, the watermark extraction procedure is going to occur. For initiating the watermark extraction process, different master shares were generated for each modified image using the same random points. Now each of these master shares underwent XOR operation with the owner share to reveal the watermark.

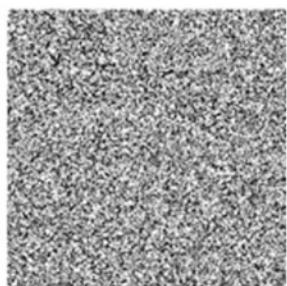
Furthermore, we have conducted experiments on the original image under the six transformations—decreased brightness, high contrast, blurry image, noise, distortion, and visible fake copyright text. After undergoing these shifts, we embedded the watermark and, later, extracted it. Figures 7, 8, 9, 10, 11, and 12 present the behavior of the original image and extracted watermark under such a scenario.



(a) Original Image



(b) Watermark



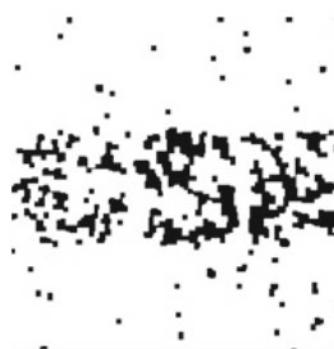
(c) Master Share



(d) Owner Share

**Fig. 6** Image watermarking

(a) Original Image with Reduced Brightness



(b) Extracted Watermark

**Fig. 7** Behavior under reduced brightness



(a) Original Image with High Contrast



(b) Extracted Watermark

**Fig. 8** Behavior under high contrast

(a) Blurred Original Image



(b) Extracted Watermark

**Fig. 9** Behavior under blur

(a) Original Image with Added Noise



(b) Extracted Watermark

**Fig. 10** Behavior under noise



(a) Original Image with Distortion



(b) Extracted Watermark

**Fig. 11** Behavior under distortion

(a) Original Image with Visible Fake Copyright Text



(b) Extracted Watermark

**Fig. 12** Behavior under visible fake copyright text

#### 4.1 Performance Parameters

We used the *matchTemplate* functionality of OpenCV to compare the obtained watermark from the six transformations with the original one. The prevalent standard of quality measurement between two images is peak signal-to-noise ratio (PSNR), represented in decibels (dB) [5]. It requires the calculation of mean square error (MSE), which is the mean of the cumulative square of the error between the reconstructed and original image. The calculation of the two metrics—MSE and PSNR—is presented in Eqs. 1 and 2, respectively. A high PSNR indicates a better quality of the reconstructed image.

Table 2 presents a performance analysis of the extracted and original watermarks under the following conditions—reduced brightness, increased contrast, blurred image, noise, distortion, and visible text. It is clear from the study that our suggested

**Table 2** Performance analysis of extracted and original watermarks

S. No.	Description	Match accuracy (in %)	PSNR (in dB)
1	Reduced brightness (Fig. 7)	61.50	47.3839
2	Increased contrast (Fig. 8)	70.62	47.8647
3	Blurred (Fig. 9)	<b>89.70</b>	<b>50.6334</b>
4	Added noise (Fig. 10)	64.64	46.0616
5	Distorted (Fig. 11)	87.54	50.0314
6	Visible text (Fig. 12)	86.76	49.7006

Bold value indicates the highest performing results

**Table 3** Comparative performance analysis of models

Watermark description	PSNR (in dB)	
	Proposed model	Adbulazeez et al. [14]
Reduced brightness	47.3839	31.78
Increased contrast	47.8647	45.26
Blurred	50.6334	48.087
Added noise	46.0616	35.25
Distorted	50.0314	47.72
Visible text	49.7006	46.18

model is handling all these types of changes well, with PSNR values ranging from 46–50 dB. The resulting PSNR values of the extracted watermarks show us that even after digital modification, the watermark still survives and is easily verifiable through this technique.

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_{(i,j)} - K_{(i,j)}]^2 \quad (1)$$

where image dimension is  $m \times n$ ,  $I$  is the original image, and  $K$  is the reconstructed image.

$$\text{PSNR} = 10 \log_{10} \left( \frac{R^2}{\text{MSE}} \right) \quad (2)$$

where  $R$  is the maximum range of values (maximum value for pixels = 255).

Table 3 presents a performance comparison of the proposed model with an approach developed by Abdulazeez et al. [14]. The authors in [14] have used the lifting wavelet transform (LWT) and singular value decomposition (SVD) on grayscale images. Further, they have incorporated a multi-objective artificial bee colony optimization for determining the multiple scaling features used against various attacks. We have tested their algorithm on our dataset for comparison. It is evident from the table that their model has not fared as expected when subjected to the images having reduced brightness and added noise. This indicates that the extracted watermarks under these two circumstances have not preserved their integrity and are not adequately verifiable.

## 5 Conclusion

The developed method successfully obtained the watermark from the carrier image without affecting it. It even allows base image modifications to simulate digital editing effects and still extract a verifiable watermark from the shares. As seen in the case of the blurred image, the highest match accuracy of 89.7% with a PSNR of 50.6 dB has been achieved. Embedding some watermarks/data in the actual image can significantly decrease the security risk. In future, we will implement methods to transform the pixels of the base image to carry additional information that acts as an invisible watermark.

## References

1. Thawkar S (2012) Digital image watermarking for copyright protection. *Int J Comput Sci Inf Technol* 3(2):3757–3760
2. Lee KH, Chiu PL (2013) Digital image sharing by diverse image media. *IEEE Trans Inf Forensics Secur* 9(1):88–98
3. Naor M, Shamir A (1995) Visual cryptography. In: Advances in cryptology-EUROCRYPT'94: workshop on the theory and application of cryptographic techniques, Perugia, Italy, 9–12 May 1994. Proceedings, vol 13. Springer, pp 1–12
4. Chandramathi S, Ramesh Kumar R, Suresh R, Harish S (2010) An overview of visual cryptography. *Int J Comput Intell Tech* 1(1):32–37
5. Yang CN (2004) New visual secret sharing schemes using probabilistic method. *Pattern Recogn Lett* 25(4):481–494
6. Hsu CS, Tu SF (2008) Digital watermarking scheme with visual cryptography. IMECS, Hong Kong, p 68
7. Abusitta AH (2012) A visual cryptography based digital image copyright protection 3(2):96–104
8. Das S, Kundu MK (2013) Effective management of medical information through ROI-lossless fragile image watermarking technique. *Comput Methods Progr Biomed* 111(3):662–675
9. Astuti YP, Rachmawanto EH, Sari CA et al (2018) Simple and secure image steganography using LSB and triple XOR operation on MSB. In: 2018 International conference on information and communications technology (ICOIACT) IEEE, pp 191–195
10. Fatahbeygi A, Tab FA (2019) A highly robust and secure image watermarking based on classification and visual cryptography. *J inf Secur Appl* 45:71–78
11. Yang CN, Lin YC, Li P (2020) Cheating immune k-out-of-n block-based progressive visual cryptography. *J Inf Secur Appl* 55:102660. <https://www.sciencedirect.com/science/article/pii/S2214212620308152>
12. Li P, Ma J, Ma Q (2020) (t, k, n) XOR-based visual cryptography scheme with essential shadows. *J Vis Commun Image Represent* 72:102911
13. Bal SN, Nayak MR, Sarkar SK (2021) On the implementation of a secured watermarking mechanism based on cryptography and bit pairs matching. *J King Saud Univ-Comput Inf Sci* 33(5):552–561
14. Abdulazeez AM, Hajy DM, Zeebaree DQ, Zebari DA (2021) Robust watermarking scheme based LWT and SVD using artificial bee colony optimization. *Indonesian J Electr Eng Comput Sci* 21(2):1218–1229
15. Ren L, Zhang D (2022) Toward privacy protection of sensed biometric features with extended visual cryptography. *Microprocess Microsyst* 91:104540
16. Devi KJ, Singh P, Thakkar HK, Kumar N (2022) Robust and secured watermarking using Ja-Fi optimization for digital image transmission in social media. *Appl Soft Comput* 131:109781

# Analytical Comparison of Deep Learning Frameworks for Semantic Segmentation with Pixel-Level Understanding



Ankit Deb Shuvrajeet Das, and Suman Deb

**Abstract** In recent years, semantic segmentation has expanded enormously. Yet, a significant number of pixel-to-pixel-level annotations are crucial to the enjoyable performance. On numerous difficult datasets for semantic segmentation, contemporary deep learning architectures produce incredibly accurate results. The state-of-the-art approaches are not readily convertible to real-time applications, since the simple adaption of such systems to cut the processing cost produces a significant drop in accuracy. The model's erroneous predictions have the potential to have a significant impact on this strategy. Visual comprehension of intricate urban street sceneries is a necessary component for a variety of applications. In the context of deep learning, object detection offers an exceptional advantage over large-scale datasets. Semantic segmentation's main issue has been model accuracy and model loss. To get grip on the problems, this paper comes with an optimized comparison of different deep learning frameworks for semantic segmentation. This paper has analyzed certain deep learning frameworks for precise semantic segmentation on a dataset. This work is pulled off using UNet, ResUNet, DeepLab V3+, and Gated SCNN, resulting in a preferable desired output. Implementing this algorithm required a deep understanding of the frameworks. The experimental result and analysis are done with different semantic segmentation frameworks on a cityscape dataset to verify the efficiency of the models. The frameworks have been compared with each other to establish reliability and accuracy.

**Keywords** Deep learning · Semantic segmentation · Convolutional neural network · Framework · Cityscape dataset

---

A. Deb · S. Das · S. Deb

Department of CSE, NIT Agartala, Paschim Barjalai 799046, India  
e-mail: [ankitdeb98@gmail.com](mailto:ankitdeb98@gmail.com)

S. Deb  
e-mail: [sumandeb.cse@nita.ac.in](mailto:sumandeb.cse@nita.ac.in)

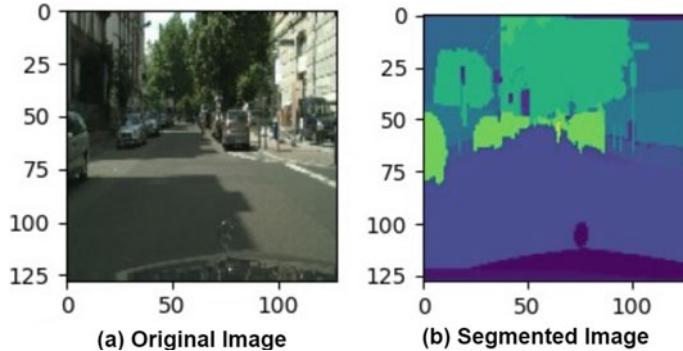
## 1 Introduction

Semantic segmentation is one of the most important topics in the field of computer vision. Semantic segmentation targets to allocate a human-defined class label to each pixel of an image, which is an effortless task in computer vision research. These semantic segmentation techniques of an image can be applied to autonomous driving [1], medical image processing, and smart cities. Semantic image segmentation has made enormous progress in recent years. In the past few years, the semantic segmentation frameworks based on deep neural networks have made incredible advancements in the field of deep learning [2]. However, most of the semantic segmentation methods are based on pixel-level annotations which are labeled manually, and it is quite expensive and also time-consuming. That does not give access to the researchers and professionals to pick the best-suited design alternative for the required task. The challenge is how to extract additional and useful training signals from the unlabeled images to allow the training of the model to generalize beyond the small labeled set. It is still quite difficult to extract particularly consistent information from remotely sensed images [3] in urban environments. On the other hand, man-made objects of the same semantic class are often built with different materials and different structures.

Most of the existing studies tackle the difficulties of supervised semantic segmentation by applying either consistency training or self-training to the data that is unlabeled. Multi-scale architecture and structured models have been introduced in different directions for the betterment of the accuracy of the architecture. The optimized deep residual network that has been used in this work is focused on the model accuracy and robustness of semantic segmentation of an image. However, little contemplation is given to the computational efficiency of the networks. By utilizing deep residual blocks, a deeper network can be constructed without having to worry about vanishing or expanding gradient issues. It also makes network training simple. Although when it comes to applications such as autonomous driving and medical image segmentation, this would have an enormous impact. Figure 1 represents the semantic segmentation of a cityscape dataset.

### 1.1 *Different Framework for Semantic Segmentation*

Conventional semantic segmentation algorithms, while considering the edges and other characteristics of the image, largely cluster pixels according to their length. There has been a major expansion and evolution of the clustering approach for semantic segmentation. This study has made use of a number of deep learning frameworks. UNet [4] is one of the most well-known and important methods for semantic segmentation. The convolutional neural network's development, known as UNet, primarily focuses on image classification. The classification of every pixel in the image guarantees that the input and output sizes of the image are the same. The building blocks



**Fig. 1** **a** The original cityscape image, **b** The segmented image

of the UNet architecture, known as ResUNet [5], are replaced by the revised residual architecture of the convolutional layer. Residual blocks are excellent at handling the vanishing gradient problem and exploding gradients, which are recent in deep neural architecture. A gated SCNN [6], a two-stream convolutional neural network configuration, is used for semantic segmentation. To eliminate noises and enable the shape stream to focus on the boundary-related information of the image, this kind of gate uses higher level activations in the classical stream. The DeepLab V3+ [7] is the expanded version of the DeepLab V3 [8]. The primary aim of DeepLab V3+ is to allocate semantic labels to every pixel of the image.

This paper is arranged into major sections describing the related work that has been done on semantic segmentation, discussion of methodology and experimental result and analysis, and finally conclusion.

## 2 Related Work

In this section of work, a variety of deep learning frameworks are discussed that has been used in semantic segmentation, mainly the deep residual network.

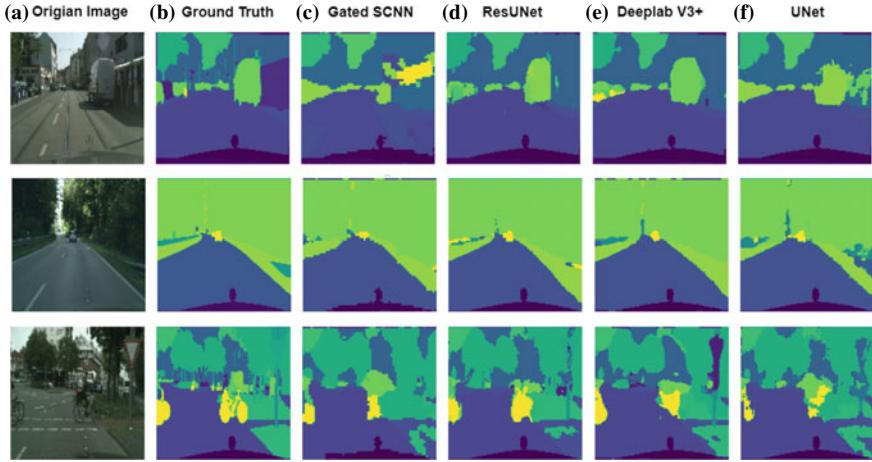
Semantic segmentation is a crucial yet challenging task to do. For each pixel in the image, predictions are made using high-level semantic characteristics. A fully convolutional network(FCN) [9] is the first semantic segmentation network that uses convolutional layers to replace the final fully-connected layer of a classification network. The approaches are based on the encoder-decoder, and the completely convolutional layer's output is smaller than the input images structure [10] that is responsible for the outputs [11] step-by-step. Although the information about the semantic segmentation has been encoded in the high-level output features, it cannot capture the long-range relationships with enhanced accuracy. The main fundamental aim of data augmentation is to regularize the learning by artificially extending the training dataset. MixUp [12] constructs a new training example by blending

**Table 1** Recent works that have been done on semantic segmentation

Existing work	Framework
Multistage attention ResUNet for semantic segmentation of fine-resolution remote sensing images [5]	Deep residual network
Swin transformer embedding UNet for remote sensing image semantic segmentation [15]	UNet
Gated SCNN: gated shape CNNs for semantic segmentation [6]	Gated shape convolutional neural network
An advanced spectral-spatial classification framework for hyperspectral imagery based on DeepLab V3+ [16]	DeepLab V3+

two images and their respective labels which are augmented. AugMix [13] applies a composition of simple transformations to original images and proposes a consistency loss-based method that will give prediction in two-augmented and one original image. Two of the main methods have been proposed in recent times, namely consistency regularization and entropy minimization. These methods do not achieve enhanced performance on their own but can be effective in conjunction with other methodologies. Physically, pixel-level annotations for semantic segmentation are very much time-consuming and costly. It is indispensable to look for all the available images to help the model learn segmentation. Consistency regularization is studied broadly for supervised segmentation. It accomplishes the predictions with various labels. The input labels augment the image randomly and urge the consistency constraints between the augmented images for the low-density region of the decision function. Self-training, self-learning, self-labeling, or decision-directed learning is originally developed for using unlabeled data in semantic classification [14]. It encompasses semantic segmentation data on the unlabeled data obtained from the segmentation model previously trained on the labeled data.

Table 1 shows the active use of different deep learning frameworks. The presented work in reference [5] shows the active use of the deep residual network in the field of semantic segmentation. In the mentioned work, ResUNet architecture has been used for the semantic segmentation of fine-resolution remote sensing images. The deep residual network is the updated version of UNet. In the work [15], it presents the use of UNet in the field of semantic segmentation. This paper also consists of gated shaped convolutional neural network and DeepLab V3+. The paper of Takikawa [6] proposed a specific wiring shape information as a separate processing branch, or shape stream, that processes input parallel to the classical stream, which is a new two-stream CNN architecture for semantic segmentation. A novel class of gates that joins the intermediary layers of the two streams is essential to this architecture. In the study of Yifan Si, a segmentation framework based on the DeepLab V3+ [7] neural network has been introduced and used it to tackle the classification issue for hyperspectral data (HSIC). The principal component analysis reduces the hyperspectral



**Fig. 2** Above figure contains **a** original image, **b** ground truth of the segmented mask, **c** segmentation using gated SCNN, **d** segmentation using ResUNet, **e** segmentation using DeepLab V3+, **f** segmentation using UNet

image's dimensionality (PCA). Spatial features are extracted using DeepLab V3+ and combined with spectral features. This paper discussed the complexities that have been faced while working with semantic segmentation frameworks such as accuracy model loss.

### 3 Methodology

With a size of  $1024 \times 2048$ , Cityscapes includes 2975 training, 500 validation, and 1525 testing photos. In good weather and during the day, images are taken from a moving car. The tests with half-resolution and full-resolution have been worked on. The model overfits the little training data, which causes the features to unduly rely on the contextual information without enough self-awareness, which is one of the reasons why features stretch too much under varied contexts. Training evolution for the deep learning family of models using conditioned versus conventional multi-task models. The regular multi-task model is shown with a dashed blue line, whereas the conditioned model is shown with a solid red line. The regular multi-task model is shown with a dashed blue line, whereas the conditioned model is shown with a solid red line. The mask exhibits greater variance during training, particularly near the point of convergence. The task is simply to reduce over-reliance on context and improve how contexts are employed. In Fig. 2, semantic segmentation has been done on the Cityscape dataset on different frameworks.

### 3.1 *The UNet Framework*

The upgraded convolutional neural network frameworks for semantic segmentation have been employed in this work. The UNet design begins with some padding applied to the input image and then performs a convolutional operation. The picture is enlarged following the transposed convolution, and it is then concatenated with the equivalent image.

### 3.2 *The ResUNet Framework*

Similar to UNet, ResUNet is made up of a bridge that connects a decoding network and an encoding network. Pre-activated residual blocks are also included in the deep residual network. The output of the encoder blocks serves as a skip connection, and it passes the input image across several blocks. The decoder creates a semantic segmentation mask using the map from the bridge and the skip connection from each encoder block.

### 3.3 *The DeepLab V3+ Framework*

The segmentation model tends to generate smooth boundaries which might not give precise accuracy for long-range objects or scenes with illegitimate boundaries. To get a grip on extracting feature maps for a comprehensive environment, DeepLab V3+ [16] was established. The model's framework is used to extract the picture feature. In the final blocks of the backbone, atrous convolution is utilized to regulate image size. Here, pooling of the atrous spatial pyramid(ASPP) network classifies the image, and after that, it goes through a 1X1 convolution to obtain the accurate size of the image with a segmented mask.

### 3.4 *The Gated SCNN Framework*

The gated-shaped convolutional neural network incorporates two steam of a network. The first steam of the network is the quality segmentation CNN, and the second steam processes information on the shape of an image in the form of segmentation boundaries. The features of the regular steam and shape steam generate a refined segmentation mask of the input image.

## 4 Metrices of the Framework

To perform semantic segmentation with good accuracy with the above-mentioned frameworks, essential matrices have to be done correctly to ensure your model performs well, these are:

### 4.1 Pixel Accuracy

Pixel accuracy [17] is a semantic segmentation metric that indicates the proportion of classified pixels in an image. This metric determines the proportion of pixels in the image that are correctly classified as all of the pixels in the image. Pixel accuracy is perhaps the easiest to understand theoretically. It is the percentage of pixels in the given image that are classified correctly. This can be shown in the Eq. 1.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

### 4.2 Intersection Over Union

IoU [18] is defined as the area of union between the predicted segmentation and the ground truth divided by the area of overlap between the two. One of the most frequently employed metrics in semantic segmentation is insertion over the union, also known as the intersection over union (IoU), or the Jaccard Index. The IoU is a relatively easy statistic that is incredibly effective. The equation for the IoU can be seen in Eq. 2.

$$\text{IoU} = \left( \frac{\text{Area of Overlap}}{\text{Area of Union}} \right) \quad (2)$$

### 4.3 F1-score

F1-score [19] is  $2 * \text{the area of overlap} / (\text{total number of pixels in both images})$ . The dice coefficient is a measurement of overlap between two masks. 1 represents a perfect overlap, whereas 0 indicates no overlap. The equation of dice coefficient can be shown in the Eq. 3. Semantic segmentation using different frameworks is shown in Fig. 2.

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

## 5 Result and Analysis

Semantic segmentation has been accomplished successfully utilizing a variety of frameworks, including DeepLab V3+, Gated SCNN, ResUNet, and UNet. ResUNet outperforms the other frameworks from this group in terms of performance. A comparison table of these frameworks has been shown in Table 2.

**Table 2** Results of different framework used for semantic segmentation

Framework	Accuracy	Dice coef.	F1-score	Iou coef.	Loss
UNet	90%	0.8616	0.9113	0.7590	0.4088
<b>ResUNet</b>	<b>93%</b>	<b>0.911</b>	<b>0.9332</b>	<b>0.8385</b>	<b>0.2548</b>
Gated SCNN	85%	0.7969	0.8570	0.6660	0.5919
DeepLab V3+	91%	0.8725	0.9137	0.7755	0.3458

The performances of the different frameworks have been discussed below:

### (i) *UNet*

Figure 3 displays the UNet’s performance. The model loss, dice coefficient, F1-score, intersection over union, and model accuracy are displayed here. Table 2 shows that UNet has a 90% model accuracy rate, a 0.8616 dice coefficient, a 0.9113 F1-score, a 0.7590 IoU coefficient, and a 0.4088 model loss.

### (ii) *ResUNet*

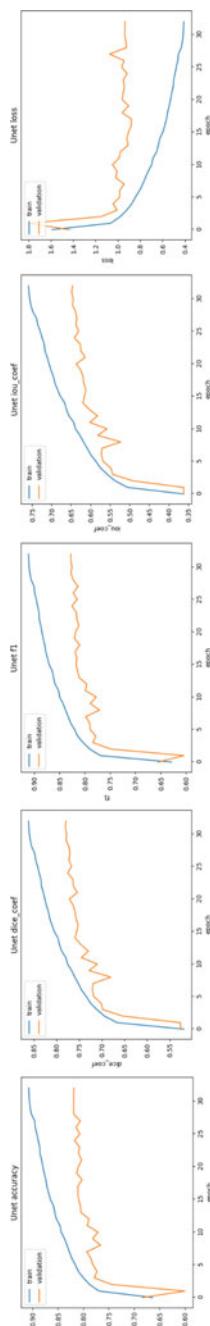
Figure 4 displays the ResUNet’s performance. The model loss, dice coefficient, F1-score, intersection over union, and model accuracy are displayed here. Table 2 makes it very evident that ResUNet has a 93% model accuracy rate, a 0.911 dice coefficient, a 0.9332 F1-score, a 0.8385 IoU coefficient, and a 0.2548 model loss.

### (iii) *Gated SCNN*

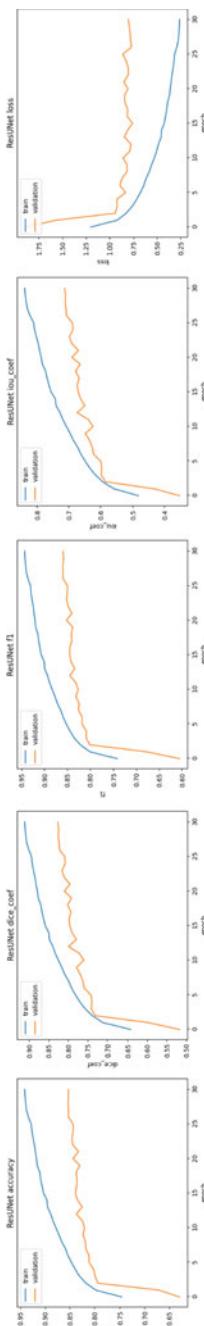
Figure 5 displays the gated SCNN’s performance. The model loss, dice coefficient, F1-score, intersection over union, and model accuracy are displayed here. Table 2 shows that gated SCNN has an 85% model accuracy rate, a 0.7969 dice coefficient, a 0.8570 F1-score, a 0.6660 IoU coefficient, and a 0.5919 model loss.

### (iv) *DeepLab V3+*

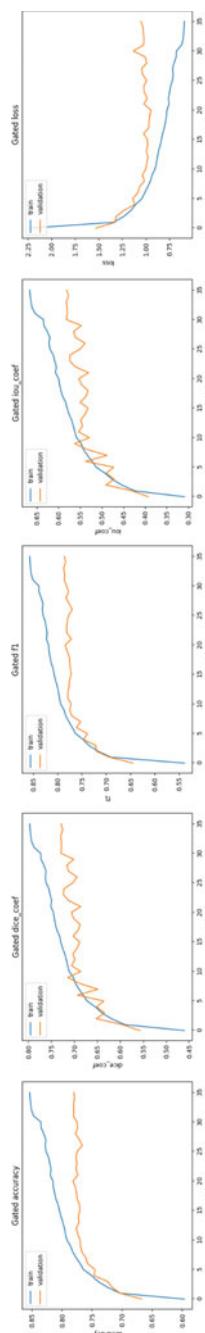
The performance of DeepLab V3+ has been shown in Fig. 6. Here, the model accuracy, dice coefficient, F1-score, intersection over union, and model loss have been shown. Table 2 shows that DeepLab V3+ is having 91% model accuracy, 0.8725 dice coefficient, 0.9137 F1-score, 0.7755 IoU coefficient, and its model loss is 0.3458.



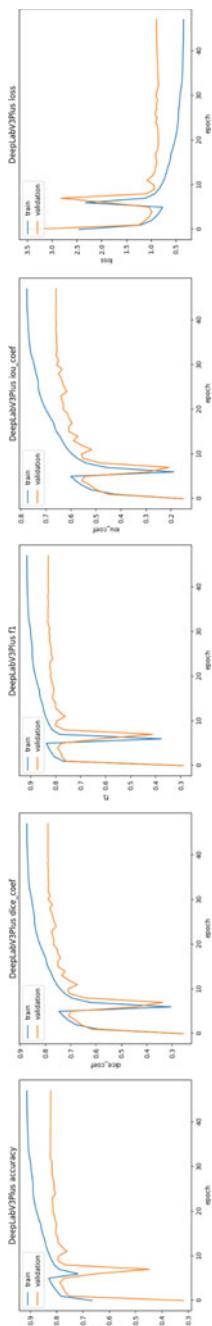
**Fig. 3** Performance of the framework UNet which contains model accuracy, dice coefficient, F1-score, intersection over union, and model loss



**Fig. 4** Performance of the framework ResUNet which contains model accuracy, dice coefficient, F1-score, intersection over union, and model loss



**Fig. 5** Performance of the framework gated SCNN which contains model accuracy, dice coefficient, F1-score, intersection over union, and model loss



**Fig. 6** Performance of the framework DeepLab V3+ which contains model accuracy, dice coefficient, F1-score, intersection over union, and model loss

## 6 Conclusion

Semantic segmentation has been successfully demonstrated on different segmentation frameworks from a dataset. Good accuracy of the required function will be important for semantic segmentation. The main objective of this paper is to segment an image with various frameworks with high accuracy and determine which is best for the dataset. It has been accomplished precisely in this regard. The training model demonstrates that the deep residual network outperforms the other frameworks in terms of accuracy. In order to quickly address the issues we encountered in this study, real-time semantic segmentation frameworks will be incorporated into further work.

## References

1. Qu Z, Jin H, Zhou Y, Yang Z, Zhang W (2021) Focus on local: detecting lane marker from bottom up via key point. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14122–14130
2. L-CCGP Florian, Hartwig Adam S (2017) Rethinking atrous convolution for semantic image segmentation. In: Conference on computer vision and pattern recognition (CVPR). IEEE/CVF, vol 6
3. Diakogiannis F, Waldner F, Caccetta P, Wu C (2020) Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J Photogramm Remote Sens* 16:94–114, 02
4. Belhaj Soulami K, Kaabouch N, Nabil Saidi M, Tamtaoui A (2021) Breast cancer: one-stage automated detection, segmentation, and classification of digital mammograms using unet model based-semantic segmentation. *Biomed Signal Process Control* 66:102481
5. Li R, Zheng S, Duan C, Jianlin S, Zhang C (2021) Multistage attention resu-net for semantic segmentation of fine-resolution remote sensing images. *IEEE Geosci Remote Sens Lett* 19:1–5
6. Takikawa T, Acuna D, Jampani V, Fidler S (2019) Gated-SCNN: gated shape CNNs for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5229–5238
7. Baheti B, Innani S, Gajre S, Talbar S (2020) Semantic scene segmentation in unstructured environment with modified deeplabv3+. *Pattern Recogn Lett* 138:223–229
8. Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. [arXiv:1706.05587](https://arxiv.org/abs/1706.05587)
9. Shelhamer E, Long J, Darrell T (2017) Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(4):640–651
10. Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: 2015 IEEE international conference on computer vision (ICCV), pp 1520–1528
11. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention (MICCAI), vol 9351 of LNCS. Springer, pp 234–241. [arXiv:1505.04597 \[cs.CV\]](https://arxiv.org/abs/1505.04597)
12. Zhang H, Cisse M, Dauphin Y, Lopez-Paz D (2018) Mixup: beyond empirical risk management. In 6th International conference on learning representations (ICLR), pp 1–13
13. Hendrycks D, Mu N, Cubuk ED, Zoph B, Gilmer J, Lakshminarayanan B (2019) Augmix: a simple data processing method to improve robustness and uncertainty. [arXiv:1912.02781](https://arxiv.org/abs/1912.02781)
14. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440

15. He X, Zhou Y, Zhao J, Zhang D, Yao R, Xue Y (2022) Swin transformer embedding u-net for remote sensing image semantic segmentation. *IEEE Trans Geosci Remote Sens* 60:1–15
16. Si Y, Gong D, Guo Y, Zhu X, Huang Q, Evans Julian, He Sailing, Sun Yaoran (2021) An advanced spectral-spatial classification framework for hyperspectral imagery based on deeplab v3+. *Appl Sci* 11(12):5703
17. Luc P, Neverova N, Couprie C, Verbeek J, LeCun Y (2017) Predicting deeper into the future of semantic segmentation. In: Proceedings of the IEEE international conference on computer vision, pp 648–657
18. Huang Y, Tang Z, Chen D, Kaixiong S, Chen C (2019) Batching soft IOU for training semantic segmentation networks. *IEEE Signal Process Lett* 27:66–70
19. Jadon S (2020). A survey of loss functions for semantic segmentation. In 2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB). IEEE, pp 1–7

# Method-Level Code Smells Detection Using Machine Learning Models



Seema Dewangan and Rajwant Singh Rao

**Abstract** Code smell detection is critical for calculating system quality and identifying issues that require more work and development. The technique of finding wrongly developed code components and implementing them is known as code smell detection. In this study, we used two method-level code smell datasets: the long parameter list and the switch statement, for detecting the code smells. A SMOTE class balancing approach is utilized to deal with the issue of class imbalance in datasets. A wrapper-based feature selection approach is used to choose the best features from each dataset. We applied three ensemble learning-based machine learning methods. To validate the model's accuracy, we utilized a fivefold cross-validation technique with five performance measurements (precision, recall, F-measure, AUC\_Score, and accuracy). Using the max voting dataset, we obtained the best accuracy of 97.12% for the long parameter list dataset.

**Keywords** Method-level code smell detection · Machine learning · SMOTE class balancing technique · Feature selection technique

## 1 Introduction

Code smells are software variations that describe the violation of fundamental planning principles like encapsulation, abstraction, module, hierarchy, and modification [1]. Code smells arise for various causes, including new developers, consumer needs, struggles, etc. Fowler et al. [2] proposed 22 informal code smells. Software engineering research describes several forms of code smells, like duplicated code, large classes, long parameter lists, and so on [2]. These smells may be used to identify refactoring opportunities [3]. As a result, detecting code smells and refactoring them are critical. Code smell detection refers to the method of recognizing code smells. The

---

S. Dewangan · R. S. Rao (✉)

Department of Computer Science and Information Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur, Chhattisgarh 495009, India

e-mail: [rajwantrao@gmail.com](mailto:rajwantrao@gmail.com)

three ways to detect code smells are metrics-based [4], rule-based [5], and machine learning-based [6].

The existing research used different machine learning (ML) models to identify code smells. Each model generates unique results. There are three leading causes for differing results: (a) The programmers can detect and find the code smells in various ways. (b) Low-detector understanding such as different techniques that recognize an unusual smell for various coding elements. (c) The detectors' threshold rates for identifying the code smells may vary.

When building software, the designer must consider functional and non-functional features to ensure the software's quality [7]. Most designers emphasize functional quality and avoid non-functional needs such as the process of reusability, evolution, maintenance effort, comprehensibility, etc., [8]. The requirement for a non-functional feature denies the software's performance while increasing complexity and software maintenance costs.

The objective of this study is to detect the code smells from two method-level code smell datasets using ML methods, deal with the issue of class imbalance, and choose the most useful metrics from each dataset using the feature selection technique (FST).

To fulfill the above objective, we developed three ML techniques with SMOTE class balancing and wrapper-based FST to detect code smells from two method-level code smell datasets. The following is an overview of our work: Sect. 2 describes the linked literature, and Sect. 3 shows the research agenda flowchart. Section 5 discusses the results and discussion, while Sect. 6 discusses the conclusion.

## 2 Literature

This section described various literature that applied different ML and FST models to detect the code smells from the source code. Fontana et al. [9] presented various ML methods on four code smell datasets: data class (DC), god class (GC), feature envy (FE), and long method (LM). They build two other code smell datasets: the long parameter list (LPL) and switch statement (SS) for future work. By applying the B-J48 Pruned technique, they achieved 99.02% accuracy for the DC dataset. Alazba et al. [10] developed seventeen ML and stacking ensemble techniques to discover code smells from six code smell datasets. They used the gain ratio FST to pick the important features from all six datasets. They obtained 92.50% accuracy for the LPL dataset using the Gaussian process (GP) approach and 88.89% accuracy for the SS dataset using the GP approach. Dewangan et al. [11] proposed six MLTs for detecting code smells across all datasets. The optimal parameters are chosen using a grid search technique. The Chi-square and wrapper-based FST favor the most important metrics from all datasets, while the logistic regression approach for the LM dataset achieves 100% accuracy. Mhawish et al. [12] demonstrated a decision tree technique to detect the code smells in datasets. To obtain the most relevant metrics from each dataset, they used two FSTs (GA-Naive Bayes and GA-CFS). They find that the highest result for the DC, GC, and LM datasets applying GA-CFS-based FST is 98.05%,

97.56%, and 94.31%, respectively, and 98.38% for the LM dataset involving GA-Naive Bayes FST. Mhawish et al. [13] used ML to detect code smells. The authors also employed GA-based FST to enhance these MLs' efficiency by identifying the best metrics from each dataset. They employed the grid search-based parameter selection approach to pick the optimum model parameters and increase the efficiency of the various MLS. Using the RF approach, they attained optimum results of 99.71 and 99.70% for the DC in the ORL\_D and REFD\_D datasets, respectively. Yadav et al. [14] introduced a decision tree technique for extracting code smell prediction principles. They employed two code smell datasets, Blob class and data class. They used the grid search hyperparameter technique to locate the code smell occurrences and identify the decision rules. They achieved the highest result of 97.62% in the data class and god class. Dewangan et al. [15] provided an ML approach for detecting code smells. They used five MLS to detect the code smells from four datasets and obtained 99.12% efficiency using the random forest approach for the FE dataset. Dewangan et al. [16] demonstrated code smell detection with dimensionally reduced MLS. They used four ML models to identify code smells and obtained 99.97% efficiency using the principle component analysis with logistic regression (PCA\_LR) approach on the DC dataset. Dewangan et al. [17] demonstrated a method of detecting code smell with the help of ensemble and deep learning models. They used SMOTE and Chi-square methods with deep learning and ensemble learning approaches. They obtained 100% efficiency using five ensemble models for the LM dataset.

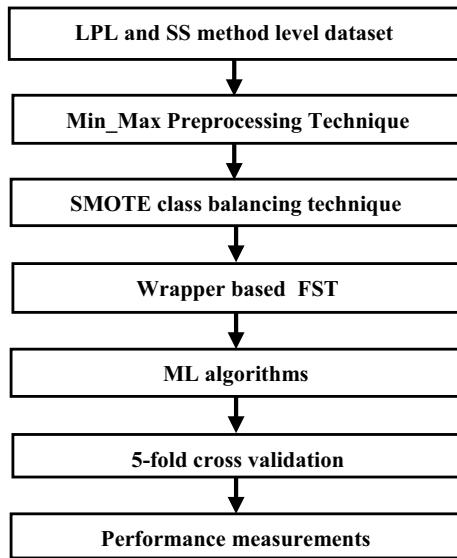
### 3 Proposed Research Plan

This section provided a strategy for identifying code smells from code smell datasets. The proposed research plan is depicted in Fig. 1 as a step-by-step process flow.

**Phase 1:** During the initial stage of our research, we made use of two datasets of code smell that were provided by Fontana et al. [9], which are a long parameter list (LPL) and a switch statement (SS). These datasets are collected from 111 systems having varying dimensions and a large number of software metrics. They looked at 55 different metrics while compiling the method-level dataset. (LPL and SS). Of the 74 systems evaluated, 37 are ineffective in detecting code smells. Visit <http://esse.re.disco.unimib.it/reverse/MLCSD.html> to get these datasets. Table 1 describes the datasets.

**Phase 2:** A pre-processing technique, min\_max, is applied to normalize the method-level code smell dataset. The min\_max technique is used to transfer all original metrics of datasets from zero to one.

**Phase 3:** According to Table 1, in both code smell datasets, the smelly sample is 140, but the non-smelly sample is 240. So, datasets have a class imbalance problem. To manage all imbalanced classes in each dataset, a synthetic minority over-sampling technique (SMOTE) class balancing technique is used. This well-known oversampling method was developed to improve random oversampling.



**Fig. 1** Proposed model

**Table 1** Dataset descriptions

Method-level dataset	Total sample	Smelly sample	Non-smelly sample	Considered metrics
LPL dataset	420	140	280	55
SS dataset	420	140	280	55

**Phase 4:** A wrapper-based FST is used to choose the important metrics from each method-level dataset. Table 2 describes the important selected metrics from each dataset.

**Phase 5:** In phase 5, we applied three ensemble learning-based ML approaches with a fivefold cross-validation method. Ensemble techniques are an ML methodology that integrates numerous single-base models to create a single best prediction model.

**Table 2** Important selected metrics

Dataset	No. of metrics	Metrics selected by Information Gain FSA
LPL	10	Method, complextype, NOP_method, package, NOLV_method, IDMMethod, NOAV_method, LOC_method, ATLD_method, is_long_parameters_list
SS	9	Method, complextype, CYCLO_method, NOAV_method, LOC_method, MAXNESTING_method, NOLV_method, IDMMethod, is_switch_statements

**Phase 6:** To examine the performance measurements (PM) of ML algorithms, we used five PMs (precision, recall, F-measure, AUC\_score, and accuracy). To evaluate the PMs, we generated a confusion matrix (CM). The CM collected the real and expected data from ML models and then calculates the precision, recall, F-measure, AUC\_Score, and accuracy.

## 4 Outcomes

To detect the code smells from two method-level code smell datasets (LPL and SS), we applied three ensemble ML models (AdaBoost, GB, and max voting). The results obtained from each ensemble ML approach (with applied wrapper-based FST, fivefold cross-validation, and the SMOTE class balancing method) are given for LPL and SS datasets in Table 3, Table 4, and Table 5, respectively. We have shown five performance parameters in each experiment table: precision, recall, F-measure, AUC\_Score, and accuracy. Table 6 compares the PM of all three ensemble methods.

From Table 3, the AdaBoost model obtained the best F-measure of 98.00% and an accuracy of 96.26% for the LPL dataset. From Table 4, the gradient boost model got the best F-measure of 97.00% and an accuracy of 96.92% for the LPL dataset. Likewise, from Table 5, the max voting model obtained the best F-measure of 97.00% and accuracy of 97.12% for the LPL dataset.

Table 6 shows the result comparison among three ML approaches on two method-level code smell datasets. We observed that the max voting model obtained the highest F-measure (97.00%), AUC\_Score (95.00%), and accuracy (97.12%) for the

**Table 3** Results of the AdaBoost classifier

Dataset	Precision (%)	Recall (%)	F-measure (%)	AUC_Score (%)	Accuracy (%)
LPL	98.00	99.00	98.00	96.00	96.26
SS	90.00	90.00	90.00	92.00	89.82

**Table 4** Results of the gradient boost classifier

Dataset	Precision (%)	Recall (%)	F-measure (%)	AUC_Score(%)	Accuracy (%)
LPL	97.00	98.00	97.00	95.00	96.92
SS	91.00	90.00	90.00	92.00	89.67

**Table 5** Results of max voting classifier

Dataset	Precision (%)	Recall (%)	F-measure (%)	AUC_Score (%)	Accuracy (%)
LPL	98.00	97.00	97.00	95.00	97.12
SS	90.00	90.00	90.00	90.00	89.88

**Table 6** Result comparison among three ML classifiers on two method-level code smell datasets

Classifier name	LPL					SS				
	P (%)	R (%)	F (%)	AUC (%)	A (%)	P (%)	R (%)	F (%)	AUC (%)	A (%)
AB	98.00	99.00	98.00	96.00	96.26	90.00	90.00	90.00	92.00	89.82
GB	97.00	98.00	97.00	95.00	96.92	91.00	90.00	90.00	92.00	89.67
MV	98.00	97.00	97.00	95.00	97.12	90.00	90.00	90.00	90.00	89.88

Where AB-AdaBoost, GB-Gradient boost, MV-Max voting, P-Precision, R-Recall, F-F-measure, AUC-Area under curve, A-Accuracy

LPL dataset, while the worst result F-measure (98.00%), AUC\_Score (96.00%), and accuracy (96.26%) obtained by the AdaBoost model for LPL dataset. The SS dataset got the best results F-measure (90.00%), AUC (90.00%), and accuracy (89.88%) using the max voting model, while the worst results obtained F-measure (90.00%), AUC (92.00%), and accuracy (89.67%) using GB model. Therefore, the max voting model is the best model to detect the code smells from the method-level code smell dataset.

#### 4.1 Result Comparison Among With and Without Applying FST

This section shows the result comparison between with and without applying FST (wrapper-based method). Table 7 shows the accuracy comparison between with applied FST and without applied FST. After comparison, we observed that after applying FST, the classifier obtained better accuracy as compared to not applying FST.

**Table 7** Result comparison between with and without applying FST

Classifier name	LPL		SS	
	Accuracy with FST (%)	Accuracy without FST (%)	Accuracy with FST (%)	Accuracy without FST (%)
AB	<b>96.26</b>	95.12	<b>89.82</b>	88.57
GB	<b>96.92</b>	95.12	<b>89.67</b>	88.88
MV	<b>97.12</b>	97.00	<b>89.88</b>	89.24

**Table 8** Result comparison between with and without applying SMOTE

Classifier name	LPL		SS	
	Accuracy with SMOTE	Accuracy without SMOTE	Accuracy with SMOTE	Accuracy without SMOTE
AB	<b>96.26</b>	94.25	<b>89.82</b>	86.57
GB	<b>96.92</b>	94.92	<b>89.67</b>	87.98
MV	<b>97.12</b>	96.60	<b>89.88</b>	87.76

## 4.2 Result Comparison Among With and Without Applying SMOTE

This section shows the result comparison between with and without applying the SMOTE class balancing method. Table 8 shows the accuracy (%) comparison between with and without applying the SMOTE. After comparison, we observed that after applying the SMOTE, the classifier obtained better accuracy as compared to without applying the SMOTE.

## 5 Result Comparison with Other Related Work

This section makes a summary of the results of our approach and how they compare to other similar efforts. Table 9 compared our findings with those of Alazba et al. [10] and Dewangan et al. [16]. Alazba et al. [10] used seventeen ML and stacking ensemble methods with a gain ratio of FST. Dewangan et al. [16] used four ML models with PCA-based FST.

- For the LPL dataset, datasets in our proposed approach, the max voting approach obtained the highest F-measure of 97.00%, AUC of 95.00%, and accuracy of 97.12%, while the Alazba et al. [10] approach obtained the highest F-measure of 92.40%, AUC of 97.00%, and accuracy of 92.50% using GP model. Likewise, Dewangan et al. [16] approach obtained the highest accuracy of 94.04% using the PCA\_LR and PCA\_RF models. They still need to present the percentage of F-measure and AUC\_Score.
- Likewise, for the SS dataset, in our proposed approach, the max voting approach obtained the highest F-measure of 90.00%, AUC of 90.00%, and accuracy of 89.88%, while the Alazba et al. [10] approach obtained the highest F-measure of 88.78%, AUC of 95.00%, and accuracy of 88.89% using GP model. Likewise, Dewangan et al. [16] approach obtained the highest accuracy of 85.72% using PCA\_KNN. They still need to present the percentage of F-measure and AUC\_Score.

**Table 9** Comparative table of our proposed approach with other related work

Year	Authors name	Method level		SS dataset					
		Best classifier		F-measure	AUC	Accuracy	Best classifier	F-measure	AUC
2021	Alazba et al. [10]	GP		92.40	97.00	92.50	GP	88.78	95.00
2022	Dewangan et al. [16]	PCA_LR, and PCA_RF	N/A	N/A	94.05	N/A	PCA_KNN	N/A	85.72
	Proposed approach	Max voting		<b>97.00</b>	<b>95.00</b>	<b>97.12</b>	<b>Max voting</b>	<b>90.00</b>	<b>90.00</b>

Where LR-Logistic regression, RF-Random forest, KNN-K-nearest neighbor

## 6 Conclusion

This study proposes three ML techniques to detect the code smell from software. Fontana et al. [9] employ two method-level code smell datasets (long parameter list and switch statement). A wrapper-based FST is utilized to identify the optimum metrics for improving model accuracy. A SMOTE class balancing technique handles the class imbalance problem in datasets.

The fundamental contribution of this study is three folds: In the first fold, three ML models (AdaBoost, GB, and max voting) are utilized. A SMOTE class balancing technique is used in the second fold, and a wrapper-based FST is operated in the third fold. In this study, we determine that the max voting model is best for both datasets, LPL, and SS. The LPL dataset obtained the best accuracy of 97.12% and for SS dataset obtained the best accuracy of 89.88%. We will use other FST and class balancing techniques in future work using different ML and deep learning algorithms.

## References

1. Booch G (1980) Object-oriented analysis and design. Addison-Wesley
2. Fowler M, Beck K, Brant J, Opdyke W, Roberts D (1999) Refactoring: improving the design of existing programs
3. Bavota G, Lucia AD, Penta MD, Oliveto R, Palomba F (2015) An experimental investigation on the innate relationship between quality and refactoring. *J Syst Softw* 107:1–14. <https://doi.org/10.1016/j.jss.2015.05.024>
4. Charalampidou S, Ampatzoglou A, Avgeriou P (2015) Size and cohesion metrics as indicators of the long method bad smell: an empirical study. In: Proceedings of the 11th International conference on predictive models and data analytics in software engineering, PROMISE '15. Association for Computing Machinery, Beijing, China, pp 1–10. <https://doi.org/10.1145/2810146.2810155>
5. Moha N, Gueheneuc Y-G, Duchien L, Le Meur A-F (2010) DECOR: a method for the specification and detection of code and design smells. *IEEE Trans Softw Eng* 36(1):20–36. <https://doi.org/10.1109/TSE.2009.50>
6. Alkharabsheh K, Crespo Y, Manso E, Taboada JA (2019) Software design smell detection: a systematic mapping study. *Softw Qual J* 27(3):1069–1148. <https://doi.org/10.1007/s11219-018-9424-8>
7. Wiegers K, Beatty J (2013) Software requirements. Pearson Education, London, UK
8. Chung L, do PLJCS (2009) On non-functional requirements in software engineering. In: Borgida AT, Chaudhri V, Giorgini P, YuE ES (eds) Conceptual modeling: foundations and applications. Lecture Notes in Computer Science. Springer, Cham, Switzerland, pp 363–379
9. Fontana FA, Mäntylä MV, Zanoni M, Marino A (2016) Comparing and experimenting machine learning techniques for code smell detection. *Empirical Softw Eng* 21(3):1143–1191
10. Alazba A, Aljamaan HI (2021) Code smell detection using feature selection and stacking ensemble: an empirical investigation. *Inf Softw Technol* 138:106648
11. Dewangan S, Rao RS, Mishra A, Gupta M (2021) A novel approach for code smell detection: an empirical study. *IEEE Access* 9:162869–162883. <https://doi.org/10.1109/ACCESS.2021.3133810>
12. Mhawish MY, Gupta M (2019) Generating code-smell prediction rules using decision tree algorithm and software metrics. *Int J Comput Sci Eng (IJCSE)* 7(5):41–48

13. Mhawish MY, Gupta M (2020) Predicting code smells and analysis of predictions: using machine learning techniques and software metrics. *J Comput Sci Technol* 35(6):1428–1445. <https://doi.org/10.1007/s11390-020-0323-7>
14. Yadav PS, Dewangan S, Rao RS (2021) Extraction of prediction rules of code smell using decision tree algorithm. In: 2021 10th International conference on internet of everything, microwave engineering, communication and networks (IMEECON), pp. 1–5. <https://doi.org/10.1109/IEMECON53809.2021.9689174>
15. Dewangan S, Rao RS (2022) Code smell detection using classification approaches. In: Udgata SK, Sethi S, Gao XZ (eds) Intelligent systems. Lecture notes in networks and systems, vol 431. Springer, Singapore. [https://doi.org/10.1007/978-981-19-0901-6\\_25](https://doi.org/10.1007/978-981-19-0901-6_25)
16. Dewangan S, Rao RS, Yadav PS (2022) Dimensionally reduction based machine learning approaches for code smells detection. In: 2022 International conference on intelligent controller and computing for smart power (ICICCS), pp 1–4. <https://doi.org/10.1109/ICICCS53532.2022.9862030>
17. Dewangan S, Rao RS, Mishra A, Gupta M (2022) Code smell detection using ensemble machine learning algorithms. *Appl Sci* 12(20):10321. <https://doi.org/10.3390/app122010321>

# Q-Learning-Based Node Scheduling for Energy Saving in WSN



Jyoti and Tamal Pal

**Abstract** Wireless Sensor Networks (WSNs), an advanced wireless network technology that is used for applications such as agriculture, health monitoring, surveillance. To operate these applications, these networks use sensor nodes that are limited in battery resources. The consumption of battery happens mainly due to communication and computation tasks performed by a sensor node. This dissipation of energy causes a short lifetime of the network. To resolve this problem, we develop a Q-learning-based node scheduling (QLNS) algorithm for saving energy resource. Firstly, we developed an algorithm to find data redundancy value and later we proposed the main algorithm QLNS to schedule the state of a node using data redundancy value and residual energy. The experimental outputs exhibited that the proposed algorithm performed better in aspects of network lifetime, average energy consumption, and the number of alive nodes.

**Keywords** Wireless sensor networks · Q-learning · Node scheduling · Data redundancy · Network lifetime

## 1 Introduction

Due to recent advancements in wireless technology, WSN has innumerable usage in the field of applications such as agriculture, medicine, environmental monitoring. In WSNs, many tiny sensor nodes are deployed over a large area to collect surrounding data. To send, receive, and process the data, a node of WSNs consumes energy. Moreover, when a node is deployed for an application, it is nearly impossible to replace or recharge the battery. The main challenge is to maintain the lifetime of a network because a sensor node runs on low battery power. In past, most of the researchers developed methods to solve the energy constrained problem in WSNs like data aggregation, low-power sensor nodes design, energy saving clustering proto-

---

Jyoti (✉) · T. Pal

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India  
e-mail: [jyoti.mail1611@gmail.com](mailto:jyoti.mail1611@gmail.com)

cols, energy-efficient routing protocols, energy harvesting, duty-cycle-based Media Access Control protocols and node state scheduling. Nowadays, due to the rise of smart applications researchers applied learning techniques to design energy-efficient protocols for these applications. It enables self-adaptive and intelligence according to the dynamics of the network. Among the existing solutions, our focus is on the node state scheduling approach and applying a learning technique with it. Scheduling of a node's state is one of the effective approach for saving energy depletion. It helps sensor nodes to keep its some module off for maximum time to save energy when it is not required to keep operating.

Many works had been proposed based on node state scheduling to maximize lifetime. They are summarized as follows. Guo et al. [1] used Q-learning-based multiple computing cooperative nodes based selection algorithm. The amount of energy consumption reduced using cooperative active nodes to complete the computation of a queue task with the event nodes. Cooperative nodes were selected using the Q-learning technique. Abadi et al. [2], combined the three new techniques that include node state scheduling, data communication restriction with Reinforcement Learning (RL) and data fusion-based packet routing. The RL-based technique helped to choose the only path that contains nodes having high residual energy during data transmission. Shreyas et al. [3] developed Energy Efficient Routing Scheme (EERS) was for duty-cycle-based sleep/awake mode of nodes. The sink performed centralized scheduling to operate a particular set of active nodes for connectivity and transmission coverage. A soft computing technique fuzzy logic was used to schedule the state of a node in [4]. Sensor nodes used attributes information from the neighbors to decide suitable action for adjusting states whenever temperature changed. Chawra et al. [5], used a memetic algorithm which was a population-based algorithm to determine nodes with an active state that satisfied coverage and connectivity constraint. It found the sleep nodes that cover a maximum number of targets. Q-learning techniques-based methods also developed in areas such as security mitigation. Subbulakshmi et al. [6] developed a markov decision process outcome prediction Q-learning algorithm. It aimed to provide security while accessing the primary spectrum. Vimal et al. [7], proposed a game-theory-based scheme that prevented attacks of jamming signal on a common channel. Q-learning had been used to determine action of players. Also Internet of Things (IoT) used Ant-colony optimization with double Q-learning to enhance the energy of IoT devices in [8]. Most of the IoT devices keep busy on sensing channel due to initial fixed network parameters to access the channel. The sensing process degrades the power of IoT devices in the network.

A WSN application needs data to be collected and process at the server. The dissemination of sensed information to the base station (BS) that directs information to the server for advanced processing requires more power than data processing and sensing. When a nodes transmits a large amount of data, they need more battery power and this causes a depletion of battery rapidly. Nodes deployed in the same area, send lots of redundant data to the sink. As a result, the redundancy problem in WSNs leads to energy consumption. So, we must eliminate the redundant data. In WSNs, as energy is a crucial resource so then Harbh et al. [9], developed an Aggregation and Transmission protocol (ATP) for reducing the consumption of data communication

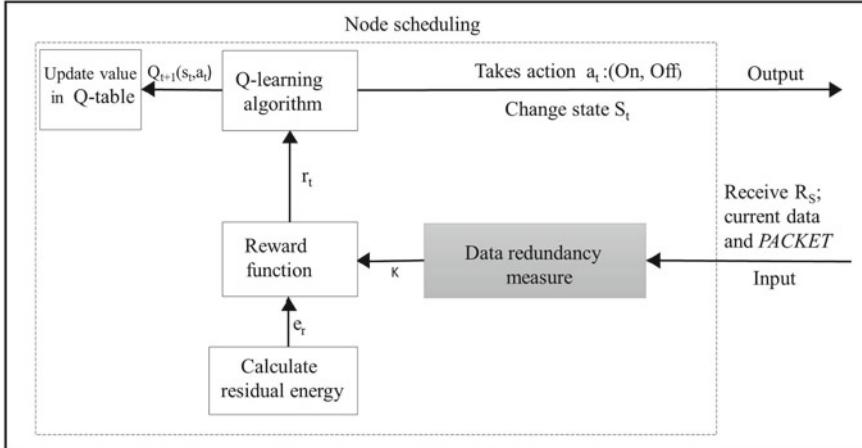
energy. It worked in two stages. The first stage was aggregation which reduced the redundancy of data sets and the second stage was the transmission. Mustafa et al. [10], used the least number of bits rather than sending whole data bits to the cluster head (CH). This method took the difference of the current reading and the last reading measures. This difference value had less number of data bits that helped in the reduction of energy consumption during communication. It has been analyzed that many works have utilized node scheduling approaches and data redundancy reduction methods to reduce energy consumption. However, the existing works suffer from several limitations such as centralized scheduling, high computation, delay, poor convergence, improper residual energy optimization, centralized data aggregation. Therefore, we propose the Q-learning-based node scheduling algorithm that considers not only residual energy but also data redundancy value to schedule the state to avoid unnecessary wastage of energy. The other part of the paper is organized as follows: Sect. 2 explains the working of the proposed work, followed by results and analysis of the proposed work in Sect. 3. Finally, Sect. 4 concludes the work with a scope for improvement in the future.

## 2 Proposed Work

In this section, we explain the algorithm developed for measuring the redundancy value and later we discuss the Q-learning-based node scheduling algorithm begin with the system model.

### 2.1 System Model

In this paper, we propose a node scheduling algorithm that makes a sensor independently learns its best action through the Q-learning technique in each operating round. A WSN comprises  $s$  number of fixed-placed sensor nodes with uniform sensing and communication range which are arbitrarily deployed in a sensing area of  $R$ . A node has neighbor nodes if the other node comes under its communication range [3]. Each sensor node denoted as  $S_i$  ( $i = 1, 2, \dots, s$ ). The number of neighbor nodes of a sensor node is  $N$  and each neighbor node of a node  $S_i$  is denoted as  $N_{ij}$  ( $j = 1, 2, \dots, N$ ). An illustration of the outline work is shown in Fig. 1. In each scheduling round, an agent node chooses an action to change its mode. If action is ON then the sensor node's state is active otherwise remains sleep for the current round. At the end of the current round, the node collects the data readings from the environment that goes to the first data redundancy measure module. In the data redundancy measure module, all the sensed data from the environment and PACKET message from neighbor nodes are collected to find the data redundancy value, i.e.,  $K$  using the data redundancy measure algorithm discussed in the next section.



**Fig. 1** The proposed model for Q-learning-based node scheduling

After finding  $K$  and residual energy  $e_r$ , these values are given to the reward function. Based upon  $K$  and  $e_r$  values, the node gets a reward. Finally, the agent updates Q-value in Q-table for the corresponding state and action using a received reward in the current scheduling round.

## 2.2 Data Redundancy Measurement

As we know, similar data are sensed by nearest neighbor nodes, which leads to redundancy of data. These redundant data add more data to the data sets. To communicate, sense, and compute, the huge data leads to unnecessary waste of energy. In the proposed work, firstly we propose an algorithm to find data redundancy value. Here data redundancy value defines how much a node has similar data from its closed nodes. The proposed algorithm to compute redundancy value  $K$  is shown in Algorithm 1. In algorithm first a sensor node  $S_i$  collects  $r$  number of readings  $R_s = \{d_1, d_2, d_3, \dots, d_r\}$  in each round. Each round consists of  $r$  reading slots and the node collects  $d_i$  reading in a slot  $i$ . The node receives  $PACKET_j$  message from its neighbor nodes  $N_{ij}$ .  $PACKET$  message contains data sets of neighbors' nodes and its ID. After receiving  $R_s$  and  $PACKET$ , the node starts to calculate redundancy of data  $DR$  as calculated in work [11]. After that, it compares the  $DR$  value with the threshold value  $\delta$ . If the value is less than or equal to  $\delta$  then the  $k$  value keeps increasing otherwise remains the same for the total number of nodes. At last, we compute the  $K$  value as the ratio of the total sum of  $k$  and the number of total adjacent nodes. The  $K$  value ranges between  $0 \leq K \leq 1$ . The reason behind this, let's assume there are two sensors  $S_1$  and  $S_2$  have  $k$  values equal to 28 out of 32( $N_1$ ) and 50 out of 100( $N_2$ ), respectively. If we consider only  $k$  value then  $S_1$  has a value of 28 less than 50 then it has a chance

to receive more reward on high redundancy which is not favorable in this work. But if we take  $K$  as a ratio of  $k$  and  $N_i$  then  $S_2$  has a value of 0.5 which is less than 0.875 which is suitable to receive a high reward.

---

**Algorithm 1:** Data redundancy measure algorithm

---

**Input:**  $PACKET_j; j \in \{1, 2, \dots, N\}$ ,  $R_S$ ,  $N_i$ ; number of neighbor nodes and  $\delta$

**Output:** Data redundancy value “ $K$ ”

**Begin**

```

1:  $k = 0$ 
2:  $Rdata_x = R_S$ 
3: for  $j \leftarrow 1$  to  $N$  do
4:    $Rdata_y \leftarrow PACKET_j$ 
5:   Calculate DR ( $Rdata_x$ ,  $Rdata_y$ )
6:   if DR ( $Rdata_x$ ,  $Rdata_y$ )  $\leq \delta$  then
7:      $k \leftarrow k+1$ 
8:   end if
9: end for
10:  $K = \frac{k}{N}$ 
11: return  $K$ 
```

**End**

---

### 2.3 Q-Learning-Based Node Scheduling

The proposed algorithm uses Q-learning which is a model-free reinforcement learning technique based on a trial and error approach. It involves an agent operating in an environment as Markov Decision Process (MDP) model with  $s \in$  states,  $r \in$  rewards(inputs) and  $a \in$  actions(outputs). The agent takes suitable action based on the policy learned from its experience. Q-learning working depends on the Q-function which calculates the Q-value of taking action in a given state. For obtaining the Q-learning function, it calculates the difference between Q-value taken in the previous state and Q-value learned in the current state’s action. Bellman-Equation is used to formulate a Q-function that tells what new Q-value is used as the Q-value for the action taken in the previous state. Q-learning agent updates the value using Eq. 1.

$$Q_n(s_c, a_c) = Q(s_c, a_c) + \alpha \{r(s_c, a_c) + \gamma \cdot \{\max_a Q(s_n, a) - Q(s_c, a_c)\}\} \quad (1)$$

In Eq. 1, at a time  $t$  agent at state  $s_t$  performs an action  $a_t$  and receives reward  $r_c$  while switching from current state  $s_c$  to next state  $s_n$ .  $\gamma$  is discount factor and  $\alpha$  is learning rate ( $0 \leq \gamma, \alpha \leq 1$ ) [12]. Each sensor node autonomously acts as an agent. We define the two states of a sensor node in set S: {Active, Sleep}. The active state is a mode in which the sensor node communicates whereas the sleep state is a mode that keeps the node’s processor off [13]. The agent performs the activity at a particular

**Table 1** Reward values

	Data redundancy value( $K$ )	Residual energy ( $e_r$ )	Values
Case I	$0.5 \leq K < 1$ (high)	$0.5e_i \leq e_r < 0.9e_i$ (high)	3
Case 2	$0.5 \leq K < 1$ (high)	$0.1e_i \leq e_r < 0.5e_i$ (low)	0
Case 3	$0 \leq K < 0.5$ (low)	$0.5e_i \leq e_r < 0.9e_i$ (high)	4
Case 4	$0 \leq K < 0.5$ (low)	$0.1e_i \leq e_r < 0.5e_i$ (low)	2

state which is called action. Action is used to schedule the state of the node. Here, we define the two actions in set A:{On, Off}. There is an action corresponding to each state of the node for scheduling. Action *On* is used to select the active state and *Off* is used to change the state to sleep mode. Each agent follows  $\epsilon$ -Greedy policy to take an action. In  $\epsilon$ -greedy, the best action is selected using probability  $1-\epsilon$  or random action using probability  $\epsilon$ . The agent aims to maximize reward by selecting the appropriate action. The agent needs to update the Q-value based on a received reward. We define a reward function hypothetically based on data redundancy value  $K$  and residual energy  $e_r$ . The reward values are assigned on different labels of  $K$  and  $e_r$  values in Table 1. The reward value is used to update Q-values and based upon it, a node schedules its state corresponding to the chosen action.

---

**Algorithm 2:** Q-learning-based node scheduling algorithm

---

**Input:** Initialize Q-table  $Q(s, a) = 0$ , discount rate  $\gamma$ ,  $\alpha$ ,  $\epsilon$  and  $t$ ; number of rounds

**Begin**

- 1: Repeat for E episodes
- 2: **for**  $i \leftarrow 1$  to  $N$  **do**
- 3:   Initialize  $e_i$
- 4: **end for**
- 5: **while** termination condition false **do**
- 6:   Choose action using  $\epsilon$ -greedy policy
- 7:   **if** action  $\leftarrow$  On **then**
- 8:     state  $\leftarrow$  Active
- 9:   **else**
- 10:     state  $\leftarrow$  Sleep
- 11:   **end if**
- 12:   Receive *PACKET* message.
- 13:    $K \leftarrow$  data redundancy measurement algorithm
- 14:   Calculate  $e_r$
- 15:   Receive reward value based on  $e_r$  and  $K$  values to update Q-value.
- 16: **end while**

**End**

---

The pseudocode of node scheduling to increase network lifetime is explained in Algorithm 2. This algorithm explains step by step procedure for a scheduling period. All the parameters are initialized and assigned initial energy  $e_i$  to every node

before rounds. At the start of each round, the node first takes an action. Based on the action, a node schedules its state. In a particular state, a node consumes some amount of energy [2]. After operating in the environment, a node receives new reading  $R_s$  and *PACKET* messages from its neighbor nodes. The algorithm then calls the data redundancy measure algorithm to calculate the  $K$  value. If a node is active, the node sends its data to the Base Station (BS) directly. After this, the node calculates its residual energy as computed in [17]. Node gets its reward value using Table 1 according to  $K$  and  $e_r$  values. After receiving a reward, the agent updates a value in Q-table. In this way, the node manages to save energy by switching its state using the proposed algorithm.

### 3 Performance Analysis

In this section, we evaluate the performance of our proposed algorithm. We compare the performance of our proposed algorithm with two existing methods. The first method is RLBEEP [2] which is based on the Q-learning method and the second method is a cluster-based energy-efficient data aggregation approach that is used to reduce data redundancy for conserving energy [14]. The evaluation of the proposed work is done from three parameters respectively which are network lifetime, average energy consumption and the number of alive nodes. The energy is calculated using the first radio model proposed in [15, 16].

#### 3.1 Simulation Environment

We build the setup on NS3, a network simulator to simulate the experiments. The parameters that are used in the experiment are shown in Table 2. We deploy 100–500 sensor nodes in the network area  $R$ . Each node can communicate with the other in its communication range. We start the experiment with nodes that have initial energy equal to 100J. Each node consumes some amount of energy which causes the depletion. An experiment is made run till any node in the network has exhausted its energy below 10% of its initial energy. There are 10 readings in a round so the total data input is number of rounds  $\times$  10.

#### 3.2 Simulation Metrics

The performance of the proposed algorithm is compared based on the parameters that include the following metrics:

**Table 2** Parameters values

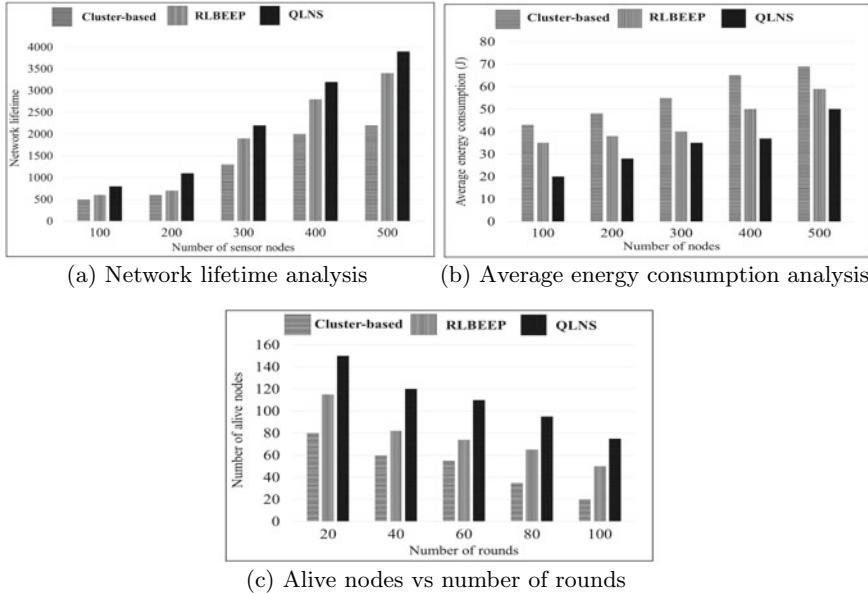
Parameters	Values
Network area ( $R$ )	100 m $\times$ 100 m
Total initial energy ( $e_i$ )	100 J
Learning parameter ( $\alpha$ )	0.8
Discount rate ( $\gamma$ )	0.9
Threshold value ( $\delta$ )	0.5
Number of slot reading ( $r$ )	10

- **Network lifetime:** It indicates the total number of cycles till all the nodes nearly have a threshold amount of energy in the network [17].
- **Average energy consumption:** It is the ratio of total energy consumed by nodes to process communication and computation to the total number of nodes in the network [3].
- **Number of alive nodes:** It is the total no of active nodes with a sufficient amount of battery resource to operate per round in the network [1].

### 3.3 Results and Discussion

The comparison between the experimental results of the proposed work with other methods is shown in Fig. 2. It shows the performance based on metric network lifetime, for that, we take 100, 200, 300, 400, and 500 nodes in the  $x$ -axis that are put against network lifetime in the  $y$ -axis shown in Fig. 2a. As the number of nodes increases in the network, the lifetime also increases it happens because whenever a node dies due to dissipation of energy the other adjacent nodes take responsible for sensing and communicating the same pattern of data to the sink. The availing of the proposed work increases lifetime by nearly 43–12% more than those of cluster-based data aggregation method and RLBEPP respectively in the case of 500 nodes.

The second metrics comparison is shown in Fig. 2b of average energy consumptions which includes the total consumption of energy by node for data communication in the network. The proposed algorithm consumes less energy than other approaches because sensor node changes its state to energy saving mode, i.e., sleep state when it covers same information from the nearest sensing region and also only active nodes communicate data to the BS. From these experiments, it can be analyzed that proposed scheme decreases energy consumption 53–42% less than other two approaches. Table 3 shows the analysis of total consumption of energy in case of active and sleep nodes of proposed method and an existing sleep scheduling mechanism. At last, Fig. 2c shows performance on metric, number of alive nodes per number of rounds of the network. As the number of rounds increases, the more number of alive nodes decreases in the network. It happens because nodes consumes energy through communicating the information. By making use of the proposed work, operating of nodes’ increases from 46 to 23%. We choose the last run of 100 for this analysis.

**Fig. 2** Results analysis of the proposed work**Table 3** Energy consumption analysis (J)

Number of nodes	Energy consumption (active)		Energy consumption (sleep)	
	QLNS	RLBEPP	QLNS	RLBEPP
10	21.5	27.4	11.2	15.3
20	25.2	31.2	12.3	14.0
30	31.3	33.5	15.1	18.2
40	38.9	42.1	16.2	22.4
50	45.6	50.1	21.4	25.8

## 4 Conclusion

In this work, we proposed the scheduling nodes using the Q-learning technique with data redundancy ratio value and residual energy. Here, we defined a reward policy that considered residual energy level and data redundancy ratio value to update the Q-value. The node changed its state to sleep mode to reduce the utilization of energy based on the Q-value. We compared the performance of the proposed algorithm with state-of-art node scheduling approaches. The experiment results shown that the proposed scheme achieved betterment in terms of network lifetime. Further, the work may be extended by considering the energy-efficient routing management, multi-cooperative working of nodes and improvement of the reward function along with the study on other performance metrics including data accuracy.

## References

1. Guo Z, Chen H (2022) A reinforcement learning-based sleep scheduling algorithm for cooperative computing in event-driven wireless sensor networks. *Ad-Hoc Netw* 130:102837
2. Abadi AFE, Asghari SA, Marvasti MB, Abaei G, Nabavi M, Savaria Y (2022) RLBEEP: reinforcement-learning-based energy efficient control and routing protocol for wireless sensor networks. *IEEE Access* 10:44123–44135
3. Shreyas J et al (2022) Energy optimization to extend network lifetime for IoT based wireless sensor networks. In: 4th international conference on smart systems and inventive technology (ICSSIT). IEEE
4. Banerjee PS et al (2022) FL-sleep: temperature adaptive multi-attribute sleep-scheduling algorithm using hesitant fuzzy logic for wireless sensor networks. *Appl Soft Comput* 123:108910
5. Chawra VK, Gupta GP (2022) Memetic algorithm based energy efficient wake-up scheduling scheme for maximizing the network lifetime, coverage and connectivity in three-dimensional wireless sensor networks. *Wirel Personal Commun* 123(2):1507–1522
6. Subbulakshmi P, Prakash M (2018) Mitigating eavesdropping by using fuzzy based MDPOP-Q learning approach and multilevel Stackelberg game theoretic approach in wireless CRN. *Cogn Syst Res* 52:853–861
7. Vimal S et al (2021) Q learning MDP approach to mitigate jamming attack using Stochastic game theory modelling with WQLA in cognitive radio networks. *J Platform Technol* 9(1):3–14
8. Vimal S et al (2020) Energy enhancement using multiobjective ant colony optimization with double Q learning algorithm for IoT based cognitive radio networks. *Comput Commun* 154:481–490
9. Harb H, Makhoul A, Couturier R, Medlej M (2015) ATP: aggregation and transmission protocol. In: 24th international conference on enabling technologies: infrastructure for collaborative enterprises, pp 134–139
10. Hammad M et al (2019) An efficient approach for representing and sending data in wireless sensor networks. *J Commun* 14(2):104–109
11. Sahar G et al (2021) Data redundancy reduction for energy-efficiency in wireless sensor networks. *IEEE Access* 9:157859–157888
12. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. *J Artif Intell Res* 4:237–285
13. Wang L, Xiao Y (2005) Energy saving mechanisms in sensor networks. In: IEEE 2nd international conference on broadband networks, pp 724–732
14. Jan SRU, Khan R, Jan MA (2021) An energy-efficient data aggregation approach for cluster-based wireless sensor networks. *Ann Telecommun* 76:321–329
15. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the IEEE 33rd annual Hawaii international conference on system sciences, p 10
16. Idrees AK, Al-Qurabat AKM (2021) Energy-efficient data transmission and aggregation protocol in periodic sensor networks based fog computing. *J Netw Syst Manage* 1–24
17. Mini S, Udgata SK, Sabat SL (2014) Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sens J* 14(3):636–644

# Performance Investigation of SVM and Modified SVM Algorithms for Acute Health Diagnosis



Pallavi Sharma, Rajni Bedi, and Vikram Dhiman

**Abstract** Early diagnosis and accurate identification of acute diseases such as heart attacks are crucial for effective control and management. Automated techniques utilizing machine learning tools can retrieve valuable information from clinical and laboratory patient data, helping physicians to maximize accuracy in identifying acute diseases. In this study, we utilized SVM, a widely-used machine learning algorithm, and its modified versions, FFA-SVM and BAT-SVM, to predict heart disease using a dataset with 14 predictive parameters. Results showed that the modified SVM algorithms outperformed the standard SVM algorithm in classification accuracy, demonstrating the effectiveness of feature selection in improving classification performance. These findings significantly impact the development of more accurate and efficient diagnostic tools in the healthcare industry. This research is essential to achieve automated, effective, timely health diagnosis, and improving patient healthcare outcomes.

**Keywords** Heart disease · Prediction · Support vector machine (SVM) · Firefly algorithm (FFA) · BAT

---

P. Sharma

Department of Electronics and Communication Engineering, Lyallpur Khalsa College Technical Campus, Jalandhar, Punjab 144001, India  
e-mail: [pallavisharma@lkcengg.edu.in](mailto:pallavisharma@lkcengg.edu.in)

R. Bedi

Department of Computer Science Engineering, Lyallpur Khalsa College Technical Campus, Jalandhar, Punjab 144001, India  
e-mail: [Rajnibedi@lkcengg.edu.in](mailto:Rajnibedi@lkcengg.edu.in)

V. Dhiman (✉)

Department of Computer Science Engineering, Gandhi Institute of Technology and Management, Visakhapatnam, Andhra Pradesh 530045, India  
e-mail: [Vdhiman@gitam.edu](mailto:Vdhiman@gitam.edu)

## 1 Introduction

Early diagnosis and accurate characterization of acute diseases, such as heart attacks, are essential for effectively managing and controlling these conditions. Machine learning techniques effectively identify patterns and relationships in clinical and laboratory data that can aid in disease diagnosis. SVM is a popular machine learning (ML) algorithm widely used for classification tasks in various fields, including healthcare. However, the performance of the standard SVM algorithm can be limited when dealing with complex and high-dimensional data. In recent years, modified versions of SVM, such as the firefly algorithm-optimized SVM (FFA-SVM) and BAT algorithm-optimized SVM (BAT-SVM), have been proposed to improve classification accuracy. In this study, we investigate the performance of SVM and two modified SVM algorithms (FFA-SVM and BAT-SVM) for acute health diagnosis using a dataset of 14 predictive parameters for heart disease. The outcome of this study showcases significant implications for developing more accurate and efficient diagnostic tools in healthcare. SVM is a powerful machine learning algorithm widely used in various medical applications, including acute disease diagnosis. However, traditional SVM may suffer from limitations in handling large datasets and may not be optimal in terms of accuracy. To overcome these limitations, modified SVM algorithms have been proposed, which optimize the SVM parameters to improve classification performance. In this study, we investigate the performance of traditional SVM and two modified SVM algorithms, FFA-SVM and BAT-SVM, for predicting acute diseases, specifically heart disease. This study aims to identify the effectiveness of these algorithms in accurately diagnosing acute diseases and to provide insights into the potential implications of utilizing machine learning algorithms for healthcare applications. At present, heart disease is one of the most dangerous issues in human health. The remedy for heart issues expressed in a review that has gotten enormous consideration in the clinical system worldwide. Heart disease is the main source of death globally. In 2019 [1], an expected 17.9 million individuals died from heart disease, addressing 32% of worldwide demises. Of these passings, 85% were because of stroke and heart attack. More than 3/4 of heart disease passings occur in low-and center pay nations. Because of non-communicable illnesses, out of the 17 million unexpected deaths, 38% were caused by heart disease [1]. Heart diseases can be predicted by addressing risk factors, for example, resting blood pressure, chest pain type, unhealthy eating regimen, and heftiness. It is vital to identify heart disease as soon as conceivable so that the proper treatment and medication can start.

Usually, standard statistical approaches and doctors' perceptions, knowledge, and involvement have been utilized for heart disease prediction. This exercise frequently results in undesired biases, errors, and high costs, as well as a negative impact on the quality of service care for patients [2]. With the rising accessibility of electronic health information, more strong and high-level computational methodologies, for example, machine learning, have become more useful to apply and find in the heart disease prediction area. One machine learning approach that has been most widely utilized is support vector machine (SVM) [3]. It offers notable properties, for

example, edge expansion and nonlinear classification through kernel tricks, and has been demonstrated to work in a wide range of real-world applications [4]. The latest research work accomplished by different researchers in the prediction of heart disease utilizing SVM is as per the following: Gao et al. [5] presented two ensemble learning techniques such as linear discriminant analysis (LDA) and principal component analysis (PCA) that were employed to increase the accuracy for the prediction of heart disease. These two techniques were utilized to choose the optimal attributes from the heart disease dataset easily. The experimental findings proved that the bagging ensemble learning technique with a decision tree (DT) accomplished higher accuracy. Pillai et al. [6] utilized a recurrent neural network (RNN), genetic algorithm, and K-mean neighbor (KNN) algorithms to foresee heart disease. Out of these algorithms, RNN attained greater accuracy. Tharwat et al. [7] proposed a BAT algorithm to optimize the parameters of SVM that were  $C$  and  $\sigma$ . Also, two other approaches were utilized: genetic algorithm (GA) and particle swarm optimization (PSO). The experimental results revealed that the BAT-SVM algorithm outperformed GA and PSO regarding classification error rate. Raju et al. [8] presented a paper for predicting heart disease utilizing the heart disease dataset. According to the experimental findings, SVM obtained a greater accuracy of 99.3% as compared to the neural network (91.1%), KNN (87.2%), and DT (82.3%). Parthiban et al. [9] utilized automatic learning algorithms like Naive Bayes (NB) and SVM to predict heart disease. The dataset of heart disease has been taken from Chennai Research Institute. The results revealed that SVM achieved a greater accuracy of (94.60%) and NB achieved an accuracy of 74%. Taie et al. [10] proposed BAT-SVM classifier-based novel model for Alzheimer's disease early detection. MRI of the brain is an important diagnostic tool for Alzheimer's disease; thus, we used it to examine regional changes in the brain that represent disease development to identify the early stages of the illness. This is a new technique that BAT solves the problem of optimizing SVM parameters to identify Alzheimer's disease using an MRI biomedical image. Yang [11] presented the BAT algorithm based on the echo location characteristics of micro-BATs and further described that the BAT algorithm used a frequency tuning method to enhance the diversity of outcomes in the population. The researchers have some glaring doubts about SVM's performance because of issues like overfitting, match-wise arrangement, and boundary regularization. Meta-heuristic algorithms can achieve an answer for such regularization by progressively calibrating the candidate arrangement and tracking down a streamlined answer for issues by upgrading the goal capability. In this paper, the parameters of traditional SVM are enhanced with the assistance of firefly algorithm (FFA) and BAT algorithms by contrasting its performance with traditional SVM by using the Eqs. (1)–(3) as follows:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (1)$$

$$V_i(t) = V_i(t-1) + (X_i(t-1) - X')f_i \quad (2)$$

$$X_i(t) = X_i(t-1) + V_i(t) \quad (3)$$

## 2 Methodology

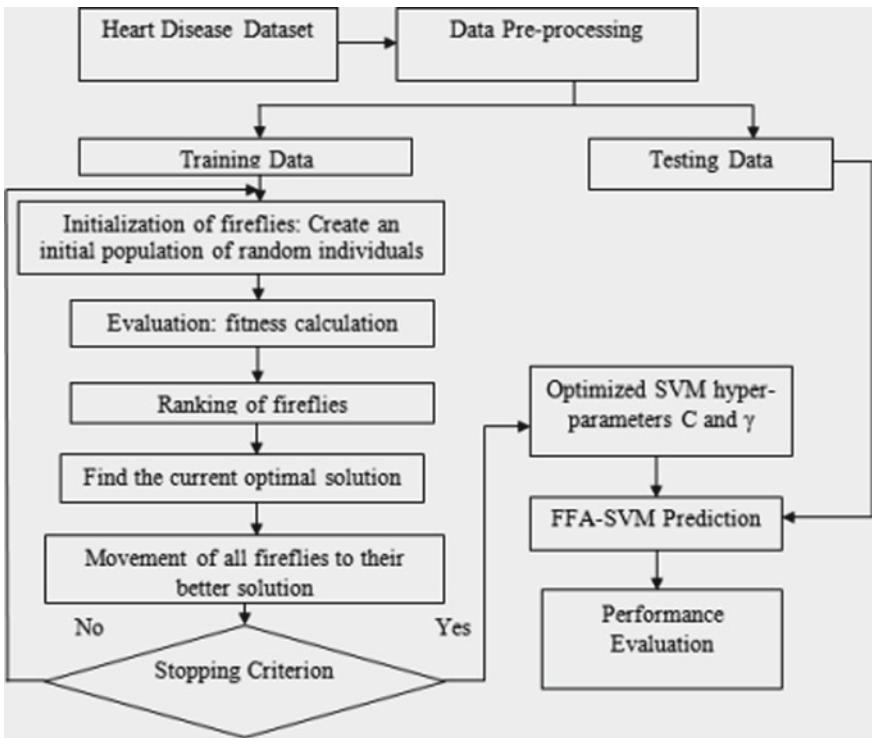
The dataset facilitating this research is based on heart disease patients taken from UCI machine learning repository. Data is provided by the Cleveland Clinic Foundation [12]. This dataset contains 13 attributes and one target column. The target column indicates 1 for heart disease and 0 for non-heart disease. The workflow representation of the methodology used in this work is shown in Fig. 1. This method is one of the Yang’s 2007 optimization algorithm [13]. This algorithm is a swarm intelligence inspired by the fireflies’ meta-heuristic approach (FFs). Because of bioluminescence, fireflies (FFs) glow up, delivering each FF a unique or closely identical brightness based on one’s attractiveness. It was revealed that two FFs could be attracted related to their brightness, which can be utilized to estimate differences in distance between the two FFs. To start the process, firstly, it is required to state the objective function  $g(y)$ ;  $y = (y_1, y_2 \dots y_d)$ . Then, a population of FFs has created  $y_i$  ( $i = 1, 2, \dots, n$ ). The brightness of each FF  $y_i$  is assessed and correlated with the selected objective function  $g(y_i)$ . A FF’s brightness is directly proportional to the objective function value. The distance  $r$  between two FFs  $i$  and  $j$  at  $y_i$  and  $y_j$  is the Cartesian distance. The distance can take any form, resulting in the best solution for various applications. The attractiveness of a FF is proportional to the amount of light visible by its neighbors. The attractiveness of a FF decreases with each movement as it moves away from its source. Because light is immersed in the medium, with  $\beta_0$  as the attractiveness at  $r = 0$ , the brightness is adjusted by using the Eq. (4) as follows:

$$\beta = \beta e^{-\gamma r} \quad (4)$$

The algorithm will be halted once the FFs have moved toward more illuminated FFs for a specific number of iterations. After that, sort the FFs by attractiveness and find the best one now [14].

### 2.1 Traditional SVM

In traditional SVM, the data points are transformed into a high-dimensional feature space. The algorithm tries to find the optimal hyperplane that maximizes the margin between the two classes. The SVM algorithm aims to find the hyperplane with the largest margin, which is expected to provide the best generalization performance on new data. The algorithm defines a boundary or a hyperplane to partition the data into classes. This study used a binary SVM classifier to classify heart and non-heart disease patients. An SVM-based classification task typically entails training and testing the SVM. Each instance in the training set has one “target value” and many “attributes.” The SVM algorithm is to discover the decision surface that augments the boundary between the instances of the two classes [15].



**Fig. 1** Workflow of FFA-SVM algorithm

## 2.2 FFA-SVM

The proposed algorithm exploits FFA concepts. In this algorithm, the SVM parameters are optimized by the FFA, and the newly acquired parameters are used as input to the SVM algorithm. The methodology of FFA combined with SVM is that the illuminating behavior of the FFs can be defined to shape an objective function, which can help with SVM parameters optimization. Here, the brightness of each FF is compared with all other FFs in the population, and one optimal position is picked depending on the most illuminated FF. In FFA-SVM, the firefly algorithm is used to optimize the parameters of SVM, such as the regularization parameter and the kernel function parameters. The methodology that is compared to the light intensity, which was typified to refresh the new places of the FFs, is looked for the best worth of the brilliance, toward which every one of the FFs moves randomly [16]. In this paper, we proposed optimizing SVM's  $C$  and  $\gamma$  parameters using the firefly algorithm (FFA-SVM). Compared to traditional SVM, FFA-SVM has several advantages. Firstly, FFA-SVM is more effective in handling nonlinear and high-dimensional data, as the firefly algorithm can help to optimize the kernel function parameters and find the optimal feature subset for the classification task. Secondly, FFA-SVM has better

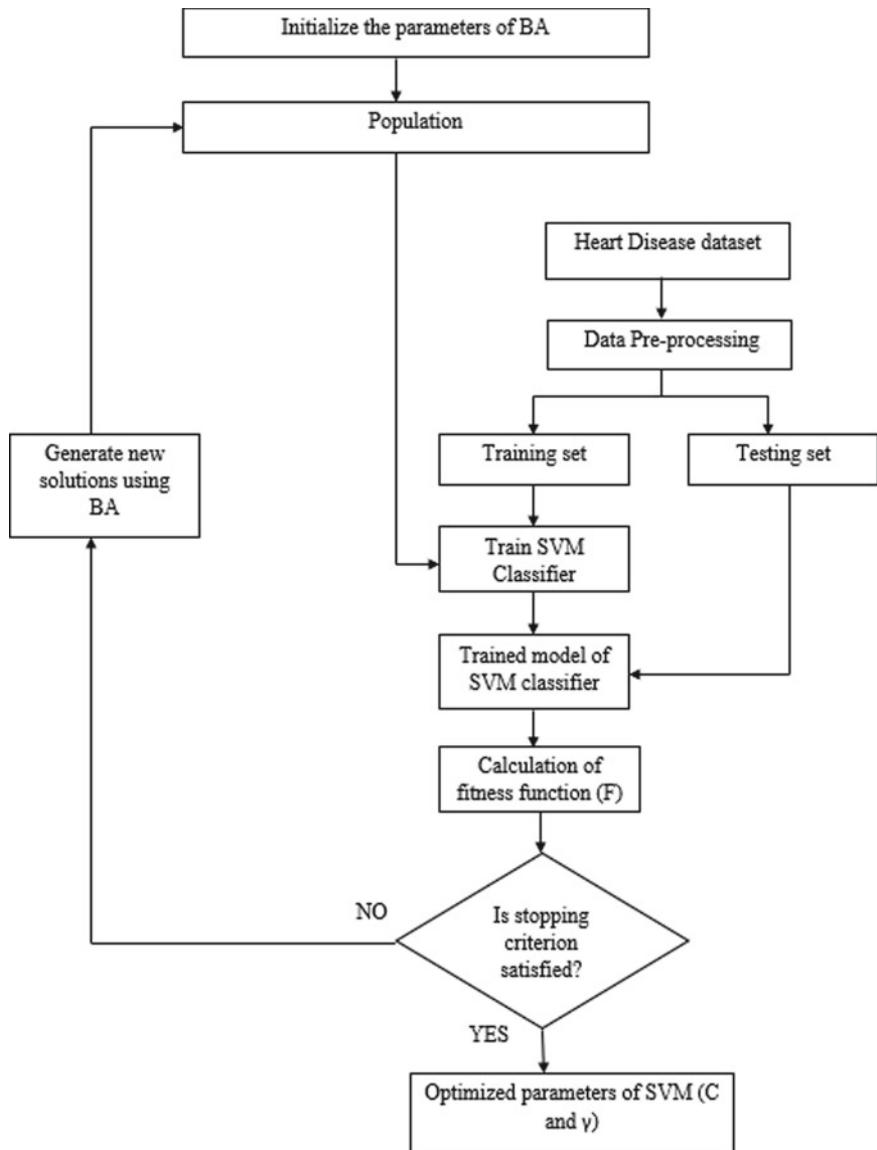
generalization performance, as the firefly algorithm can help to avoid overfitting by optimizing the regularization parameter. Lastly, FFA-SVM has a faster convergence rate, as the firefly algorithm can quickly converge to the optimal solution. Overall, FFA-SVM is a promising approach for classification tasks that require high accuracy and robustness. However, FFA-SVM also has some limitations, such as the sensitivity to the initial parameter values and the need for extensive computational resources.

### 2.3 BAT-SVM

BAT is a swarm algorithm that performs searches using a population of agents. For SVM parameter selection, BAT will search for the best  $C$  and  $\gamma$  based on the accuracy of SVM. BAT algorithm-support vector machine (BAT-SVM) is another modified version of traditional SVM that utilizes the BAT optimization technique to enhance the performance of SVM in classification tasks. Compared to traditional SVM, BAT-SVM has several advantages. Firstly, BAT-SVM is more effective in handling nonlinear and high-dimensional data, as the BAT algorithm can help to optimize the kernel function parameters and find the optimal feature subset for the classification task. Secondly, BAT-SVM has better generalization performance, as the BAT algorithm can help to avoid overfitting by optimizing the regularization parameter. Lastly, BAT-SVM has a faster convergence rate, as the BAT algorithm can quickly converge to the optimal solution. Overall, FFA-SVM, BAT-SVM, and other modified SVM algorithms demonstrate the potential for improving the performance of traditional SVM in handling complex and high-dimensional data. The key parameters of the BAT algorithm are as follows: Initially, the population size is set at 100:20. The second option is the number of generations, which might be between 1 and 100. The minimum frequency  $f_{\min}$  is 0, and the  $f_{\max}$  is 2. The tweaking of the parameters for the optimization algorithms and choosing BAT algorithm parameters has been demonstrated to be very significant regarding execution. The workflow of the BAT-SVM algorithm is shown in Fig. 2.

## 3 Results and Discussion

The UCI dataset collected was analyzed using the methodology described in Sect. 2. The heart disease dataset has 14 features, which include 13 features and one target class, and the 14th feature implies the heart disease classification or status. The description of the dataset of heart disease is given in Table 1. This work used data pre-processing to perform data cleaning. Firstly, the attributes were scaled to be in the same range. In the second pre-processing step, the categorical data with different categories were converted to numerical data so the classifier could easily understand and use it for training. For training both models, 80% of features were used for training, and 20% were used for testing.



**Fig. 2** Workflow of BAT-SVM algorithm

The parameter settings of the traditional SVM, BAT-SVM, and FFA-SVM determined after preliminary experimentation are shown in Table 2. The study compared traditional SVM, FFA-SVM, and BAT-SVM performance measures in predicting heart disease. The results showed that FFA-SVM outperformed traditional SVM and

**Table 1** Description of heart disease dataset

S. No.	Attribute	Representation
<b>1</b>	<b>X_Age</b>	<b>Age</b>
2	X_Sex	Sex
<b>3</b>	<b>X_Chestpain type</b>	<b>CP</b>
4	X_RestBloodPressure	RestBP
<b>5</b>	<b>X_Serumcholesterol</b>	<b>Chol</b>
6	X_FastingBloodSugar	FBS
<b>7</b>	<b>X_ResElectrocardiographic</b>	<b>RestECG</b>
8	X_MaxHeartRate	Thalach
<b>9</b>	<b>X_ExerciseInduced</b>	<b>Exang</b>
10	X_Oldpeak	Oldpeak
<b>11</b>	<b>X_MajorVessels</b>	<b>CA</b>
12	X_Slope	Slope
<b>13</b>	<b>X_Thal</b>	<b>Thal</b>
14	X_Class	Class

BAT-SVM regarding various performance measures, including accuracy, F1-score, precision, sensitivity, and specificity.

The accuracy of FFA-SVM was 88.5%, while the accuracy of traditional SVM and BAT-SVM was 80.3% and 86.9%, respectively. Similarly, FFA-SVM had a higher F1-score of 89.9% compared to traditional SVM and BAT-SVM, which had F1-scores of 82.9% and 88.2%, respectively. Moreover, the precision of FFA-SVM was

**Table 2** Parameters of traditional and proposed work

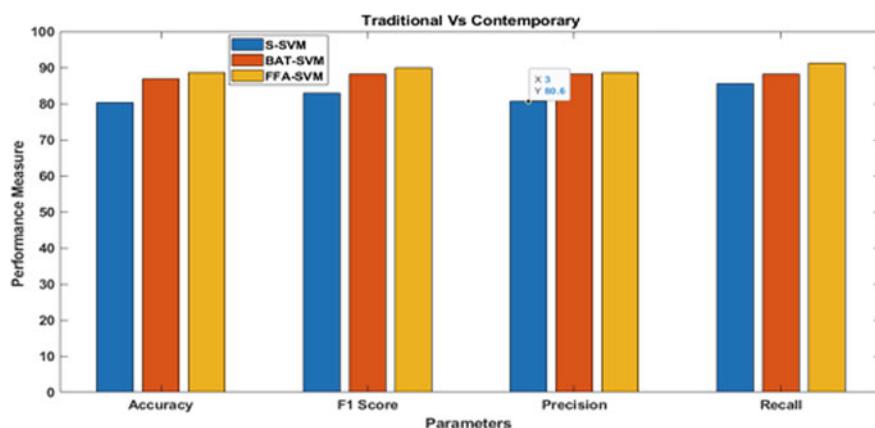
Parameters	Traditional SVM	BAT-SVM	FFA-SVM
<b>Classifier</b>	<b>SVM</b>	<b>SVM</b>	<b>SVM</b>
Kernel function	rbf	Rbf	rbf
<b>Iterations For FFA</b>	—	<b>100</b>	<b>50</b>
C	0.1	999.9955	593.7715
<b>Γ</b>	<b>Scale</b>	<b>0.0014907</b>	<b>0.000109</b>
Lower and upper bounds of the population for FFA (C)	—	—	0.1, 1000
<b>Lower and upper bounds of the population for FFA (<math>\gamma</math>)</b>	—		<b>0.0001, 1</b>
Dimensional search space (D)	—	2	2
<b>Loudness (A)</b>	—	<b>0.50</b>	—
Pulse rate (R)	—	0.50	—
<b>Minimum frequency (<math>F_{\min}</math>)</b>	—	<b>0.00</b>	—
Maximum frequency ( $F_{\max}$ )	—	2.00	—

**Table 3** Comparison of traditional SVM, BAT-SVM, and FFA-SVM using performance measures

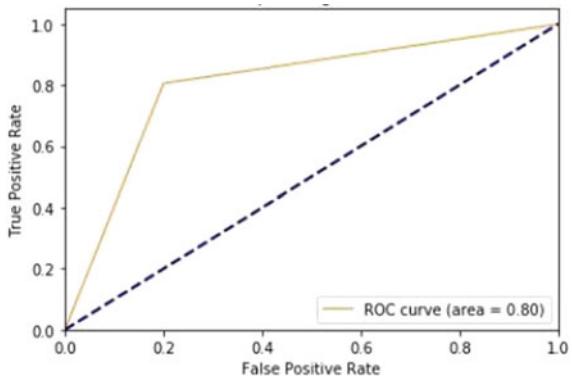
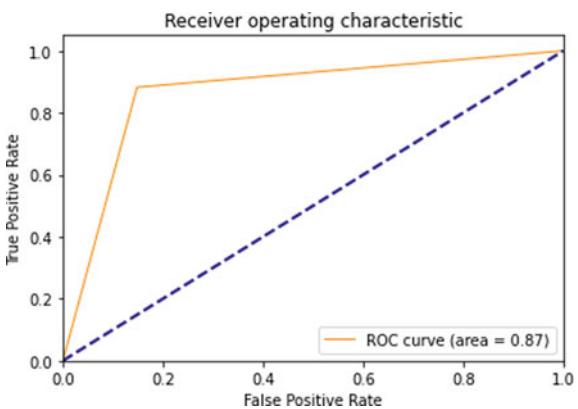
Performance measures	Traditional SVM (%)	BAT-SVM (%)	FFA-SVM (%)
<b>Accuracy</b>	<b>80.3</b>	<b>86.9</b>	<b>88.5</b>
F1-score	82.9	88.2	89.9
<b>Precision</b>	<b>80.6</b>	<b>88.2</b>	<b>88.6</b>
<b>Recall (sensitivity)</b>	<b>85.53</b>	<b>88.2</b>	<b>91.2</b>
Specificity	74.1	85.2	85.2

88.6%, while the precision of traditional SVM and BAT-SVM was 80.6% and 88.2%, respectively. Additionally, the sensitivity of FFA-SVM was 91.2%, higher than that of traditional SVM (85.53%) and BAT-SVM (88.2%). These findings indicate that FFA-SVM is a more effective and accurate algorithm for predicting heart disease than traditional SVM and BAT-SVM. The better performance of FFA-SVM can be attributed to its ability to optimize the SVM parameters more efficiently, leading to better feature selection and more accurate classification results. The study results have significant implications for developing more efficient and accurate automated tools for diagnosing acute diseases, which could ultimately improve healthcare outcomes for patients. Comparison of the performance measures, as shown in Table 3, revealed that the FFA-SVM model performed better than traditional SVM and BAT-SVM in terms of accuracy (88.5%), F1-score (89.9%), precision (88.6%), sensitivity (91.2%), and specificity (85.2%). Figure 3 shows the graphical comparison of traditional and contemporary models.

The graphical representation of the traditional SVM model and FFA-SVM in terms of the ROC curve is shown in Fig. 4 and Fig. 5, respectively. This curve depicts the number of correct positive classifications achieved with an increase in the rate of false



**Fig. 3** Comparison of traditional SVM versus BAT-SVM versus FFA-SVM

**Fig. 4** ROC SVM**Fig. 5** ROC FFA-SVM

positives. The study compared the performance of traditional SVM, BAT-SVM, and FFA-SVM in predicting heart disease using a dataset with 14 predictive parameters.

## 4 Conclusion

This paper described using traditional SVM, BAT-SVM, and FFA-SVM in the heart disease classification process. When SVM optimized with FFA, then it has attained an accuracy of 88.5%, F1-score of 89.9%, precision of 88.6%, sensitivity of 91.2%, and specificity of 85.2%, respectively. Therefore, heart disease prediction through the FFA-SVM algorithm outperformed the traditional SVM and BAT-SVM approach. More datasets, feature extraction, other classifier algorithms, and parameter tuning in classifiers will be included in future work. Overall, FFA-SVM is a promising approach for classification tasks that require high accuracy and robustness. However, FFA-SVM also has some limitations, such as the sensitivity to the initial parameter values and the need for extensive computational resources.

## References

1. World Health Organization (WHO): [https://www.who.int/news-room/fact-sheets/detail/cardio-vascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardio-vascular-diseases-(cvds)). Last accessed 1 May 2022
2. Palaniappan S, Awang R (2008) Intelligent heart disease prediction system using data mining techniques. In 2008 IEEE/ACS International conference on computer systems and applications. IEEE, pp 108–115
3. Cortes C, Vapnik V (1995) Support vector machine. *Mach Learn* 20(3):273–297
4. Moguerza JM, Muñoz A (2006) Support vector machines with applications. *Stat Sci* 21(3):322–336
5. Gao X-Y et al (2021) Improving the accuracy for analyzing heart diseases prediction based on the ensemble method. *Complexity* 2021:1–10
6. Sowri Raja Pillai N, Kamurunissa Bee K, Kiruthika J (2019) Prediction of heart disease using RNN algorithm. *Int Res J Eng Technol* 6(3):4452–4458
7. Tharwat A, Hassaniene AE, Elnaghi BE (2017) A BA-based algorithm for parameter optimization of support vector machine. *Pattern Recogn Lett* 93:13–22
8. Raju C et al (2018) A survey on predicting heart disease using data mining techniques. In 2018 Conference on emerging devices and smart systems (ICEDSS). IEEE, pp 253–255
9. Parthiban G, Srivatsa SK (2012) Applying machine learning methods in diagnosing heart disease for diabetic patients. *Int J Appl Inf Syst* 3(7):25–30
10. Taie SA, Ghonaim W (2021) A new model for early diagnosis of Alzheimer's disease based on BAT-SVM classifier. *Bull Electr Eng Inform* 10(2):759–766
11. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010), pp 65–74
12. Dataset UH (2022) UCI machine learning repository [online]. <https://archive.ics.uci.edu/ml/machine-learning-databases/heartdisease/Heart>. Last accessed 20 May 2022
13. Yang X-S (2009) Firefly algorithms for multimodal optimization. In Stochastic algorithms: foundations and applications: 5th International symposium, SAGA 2009, Sapporo, Japan, October 26–28, 2009. Proceedings 5. Springer Berlin Heidelberg
14. Yang Y, Liu X (1999) A re-examination of text categorization methods. In Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp 42–49
15. Zhang T, Oles FJ (2001) Text categorization based on regularized linear classification methods. *Inf Retrieval* 4:5–31
16. Sharma A et al (2013) Optimization of SVM classifier using Firefly algorithm. In 2013 IEEE Second international conference on image information processing (ICIIP-2013), pp 198–202

# An Efficient Multifactor Authentication System



Shreya Verma, Mansi Singh, Krittika Chaturvedi, and B. K. Tripathy

**Abstract** Most typical authentication mechanisms in this digitized world use a user-name and password to protect information. Two-factor authentication (2FA) mechanisms were introduced to add an extra layer of security. As 2FA is vulnerable to advanced attacks such as real-time replay, multifactor authentications (MFA) are being introduced and require multiple factors or proofs of identity during authentication. Thus, herein, we propose a multifactor authentication methodology using text-based encryption with a self-generated algorithm-Twine and image encryption with GENETIC algorithms. In our approach, when a user attempts to register on a Web page, they must enter details along with essential user id and password, further taking them to an image selection page where they can choose the image of their choice as a second layer of the authentication protocol. Hence, when a user attempts a login into the portal, they can be provided with options to enter via a graphical or text-based password depending on the feasibility and usage of the Web page. The introduction of a lightweight cryptographic algorithm here ensures an improbable zone for machine guessing strategies, and an innovative component for the user's password, since both graphical and textual, will be hashed and stored and will have no permanent space in the database. It is observed that graphical passwords are easier to use and remembrance. Hence, this could be the next revolution in admin panels for multitude of security driven purposes.

**Keywords** Multifactor authentication · Encryption · Twine · Genetic algorithm · Graphical password · Text-based password

---

S. Verma · M. Singh · K. Chaturvedi · B. K. Tripathy (✉)  
School of Information Technology and Engineering, VIT, Vellore, India  
e-mail: [tripathybk@vit.ac.in](mailto:tripathybk@vit.ac.in)

S. Verma  
e-mail: [shreya.verma2019@vitstudent.ac.in](mailto:shreya.verma2019@vitstudent.ac.in)

M. Singh  
e-mail: [mansidhananjay.singh2019@vitstudent.ac.in](mailto:mansidhananjay.singh2019@vitstudent.ac.in)

K. Chaturvedi  
e-mail: [krittika.chaturvedi2019@vitstudent.ac.in](mailto:krittika.chaturvedi2019@vitstudent.ac.in)

## 1 Introduction

Due to massive explorations in Internet technologies, security and authentication issues become an important challenge in all spheres of our life. The services provided efficiently around the globe have been grossly affected due to the ever increase in the numbers of consumers and the smart devices. Thus, to provide better security for someone's account, the idea of authentication has been introduced. In the authentication process, it is supposed that some secured piece of information provided by a system or an individual is to be processed to verify their genuineness as an user of the system. It has been a practice to use a single piece of information for this purpose. Keeping the ease in use and familiarity of the users, a simple single-factor authentication (SFA) is preferred. In spite of the advantages as above, it has a major drawback. If the knowledge of the password collected by mistake or through acquaintance with the user, then any adversary can enter into the system. This has necessitated the development of multifactor authentication (MFA), which takes care of higher level of security and protection of systems and any other protected services or better methods to follow and can keep ay bay unauthorized access. In this work, we propose the use of graphics (images) to replace the traditional use of alphanumeric texts for authentication. The requirement is that the user has to select a part of an image provided as password. Most of the times recalling a saved portion of a familiar image seen earlier seems easier than recalling some texts, which are restricted to be not matching with any common string, which are unlimited in size. Moreover, it embeds the strength of encryption for the authentication measures through Twine, a novel encryption algorithm which is simple in complexity for encoding as the attacker has to break the barriers of 23-round Twine-80 and 24-round Twine-128. Thus, we have implemented 36 rounds which provide a sufficient buffer. Our scope of study involves all of this being integrated with the genetic algorithm, so as to inculcate the randomness involved in the crossover and mutation processes of the genetic algorithm [1] so as to generate the required hash for the authentication process. The randomly generated data is transformed into the encoded form by applying these processes, making it quite robust and hard to crack. This way, MFA can be the vital method to assure the sustainability of online interactions and secured transactions while complicating the utilization of any system.

## 2 Literature Review

A coalition of one-time secure password and image authentication tests [2, 3] generated from a grid interface provided through 3DES conversion and cued click points was proposed in [4] where the user selects the registered click points during the password and set of images' creation. Results showed that graphical authentication improved efficiency and scalability but users must recollect the exact points which are challenging [5]. In [6], a comparative study between 2 and 3D layouts in GUI

interface is made to create a graphical password. For users, selecting images in 3D layout through navigation requires less time than usual 2D layouts in deciding their password. This portrayed a 3D graphical authentication layout. But, the password strength remained undetermined here. In [7], the security approaches of image-based password authentication and hash-MAC-based OTP are combined. The hash-MAC-based OTP generates a secure one-time password using the SHA-1 method. A novel scheme based on the knowledge-based user authentication; called the “Pict-Place authentication” (PPA); was proposed in [8] which was used to improve the memorability of the credentials. It is observed from the results that reduction of time for operation and improve in security can be obtained using PPA. A password hiding protocol that uses LSB image steganography and data hiding technique for image-based authentication was done in [9]. The initial phase included a registration module while applying one-time pad followed by columnar transposition and finally image partitioning and embedding is done. A photo response non-uniformity (PRNU)-based authentication allowing a participant to join a Zoom conference was done first in [10]. If this is not the case, password authentication is used. The direct participation of participant is not required in PRNU because he/she is not requested to supply his/her biometric, input a password, or answer a security question, making it seamless. A validation system that identifies images in a particular order where bank requires that entering the required information registration is to be done by the user through a GUI was proposed in [11]. The encoding of pixels each of three bits is carried out for 24-bit image bytes to enhance security. Through this, ELSB and LSB are implemented such that the pattern gets embedded into the image using ELSB algorithm. In [12], a survey made explains the user ease in selecting passwords where users prefer images and emojis as their passwords. Around 30 users that were investigated revealed how image height affects the memory of graphical password. The simpler the password is, the easier it is to remember. A drawback revealed that multiple options randomize the choices of a user; hence, a user gets irritated. A 2-stage registration process in [13] has the user input as a unique pin code and a 500 cue point sliced image that generates a hash code. The later process is repeated five times. This study also revealed that more than 50% of users seemed to remember their password taking an edge over textual passwords. In [14], two techniques using passwords-based on graphics were used to take care of shoulder surfing attacks. Two prototypes were used for a comparative analysis of surveys based on response collected from users. User has to draw the same strokes during logging in which was used by him/her during registration. The satisfactory rate was 91% with both methods.

The algorithm set used in [15] proceeds with textual followed by image-based, however, instead of slicing of images, it follows random sets to authenticate, and the user must find the password from the random sets that the user themselves uploaded. However, the system has drawbacks like shoulder surfing, attack prone, and possible eavesdropping. To mitigate, they provided a zero-knowledge-based protocol during image-based recognition authentication. In [16], a mechanism to fight against cryptanalysis is provided. It uses an S-box for generation of dynamic passwords and inputs images in random order creating a 2-layer authentication system. Through this successful approach, brute-force attack, man-in-the-middle attack, dictionary

attack, and replay attack could be tackled. However, the application of this algorithm is on a minuscule level; hence, scalability is an issue. Strength of the created passwords as well as the relationship between visual behaviors using the FD eye tracking data and FI individuals to create a GUA help-based approach to the saliency mask were considered in [17]. PassBYOP, a scheme for public terminals in shaping a fresh graphical password system replacing the typical digital images which are static with personalized physical tokens, was proposed in [18]. It displayed digital pictures on a mobile phone like physical user-owned device. A system camera is presented with the images of the users followed by the password in the form of the live videos of the token sequentially. Composing a password from malware or shoulder surfing or camera-based observation was discouraged through the above results. An authentication method to hide sensitive data under a cover image, preventing an unauthorized user from accessing sensitive data, was proposed in [19]. The registration action appears when a user launches the application for the first time. For 3 attempts, the application directs the unauthorized user to enter the username and password. This yields high PSNR and improves the low MSE value quality of steganography. An interface by GPAS was provided in [20] where the user was introduced to a scene, a human character, and some objects. In [21], 3 stages of identity verification are included: textual password, bot attack authentication, and color code detection. There are five colors included during registration, and the user must choose three in a specific order. The user will then enter the same code in the same order while logging in. Since the order of the colors is always unpredictable, the user must remember the code; hit and try. This will simply not work as there are 120 different color combinations in this security check. A movable frame with the set of text and image-based grid for user authentication was done in [22]. During registration, the user is shown 16 random images, and the user is asked to select 5 images to set as a password. The graphical and text-based authentication methods were compared based on reduced cognitive load, shoulder surfing prevention, and average time for login. Both hand geometry and air hand writing were used to propose a multifactor authentication system in [23]. After receiving a request for authentication, the following algorithms are run by the server: (a) Signal matching algorithm in which auxiliary information and stored signal is used to match with motion signal (b) Hand geometry matching matches hand geometry  $h$  with stored hand geometry template and showcased a high false non-match rate equal to false match rate. In [24], a protocol which uses authentication procedure on multiple factors was developed basing upon secret sharing for smart home environments and elliptic curve cryptography (ECC). The authentication process requires only a face image of the user. For forensically linking images to digital cameras, PRNU is used with three phases of user registration, mutual authentication, and PRNU update. The two shares are kept by the pair of entities, and both shares are required to reconstruct the secret.

### 3 Experimental Setup

This section provides description of the model and the overall architecture of system.

#### 3.1 Module Description

The proposed authentication consists of two major modules:

- **Registration Module:** The user is prompted initially to enter the textual password that is encrypted using the Twine encryption algorithm. This encrypted password is hashed and stored in the database. The user is then prompted to select the image and click the grid points in sequence for genetic password authentication. The selected image and click grid points are encrypted using the genetic algorithm and hashed to store the hashes in the database.
- **Login Module:** The user is prompted to enter the textual password that is encrypted using twine, and its hash value is compared in the database. After the textual authentication, the user is prompted to go through three rounds of image selection that is encrypted using the same genetic algorithm, and the hash is compared in the database. If the hash is the same, then the second authentication process completes, and the dashboard is displayed.

#### 3.2 System Architecture

Figure 1 describes the road map for the multi-stage user authentication process. It enumerates generation of the image if the username is valid and found in the database (as seen earlier) and if the generated hash values are matching depending on the condition of the graphical password coming out to be of a matching set or net. If all the conditions are satisfied, only then one gains access to the sensitive information/critical application.

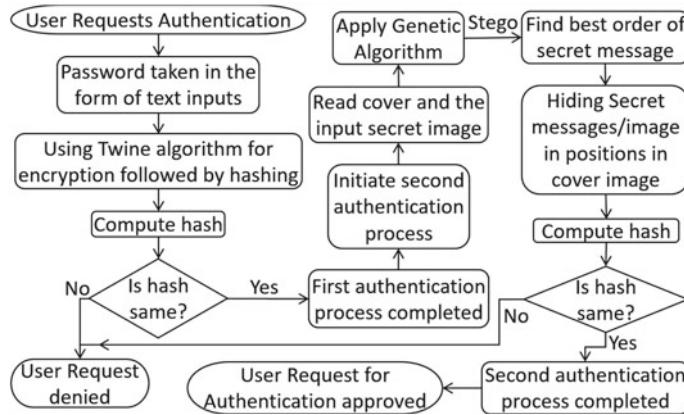
#### 3.3 Pseudo Code

The following is the pseudo code.

```

1.Start:
2.Input: sername, userPassword, reEnteredPassword
3.While userPassword != reEnteredPassword
4.    Input: reEnteredPassword
5.EndWhile
6.If (userPassword == reEnteredPassword)
7.    encryptedPassword = twineEncryptionAlgorithm(userPassword)

```



**Fig. 1** System architecture

```

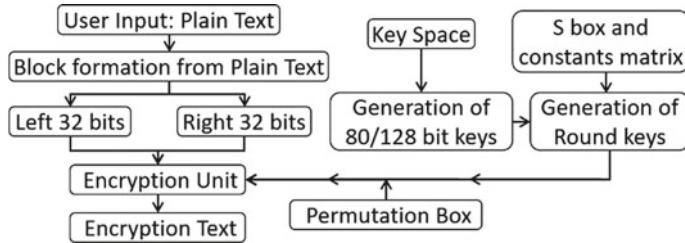
8.     hashedPassword = SHA-256(encryptedPassword)
9.     Store hashedPassword in the database
10.    EndIf
11.    For i = 0 to 4
12.      Input: userImage
13.      Slice Image
14.      Input: userSelectCuePoints
15.      encryptedCuePoints = GeneticEncryptionAlgorithm(userSelectCuePoints)
16.      hashedCuePoints = SHA-256(encryptedCuePoints)
17.    EndFor
18.    Store hashedCuePoints in the database
19. End
  
```

## 4 Detailed Methodology

This section provides detailed description of the proposed methodology.

### 4.1 Text-Based Encryption—Twine

- Twine is a 64-bit block cipher that comes in 2 variants: Twine-80 and Twine-128 using 80-bit and 128-bit keys, respectively, derived from generic Feistel structure (GFS).
- It divides a 64-bit block of plain text into 16 4-bit sub-blocks for processing using a 4-bit S-box and a permutation box to perform diffusion. Main aspect in which Twine differs from Feistel architecture is GFS simply uses cyclic shift, whereas



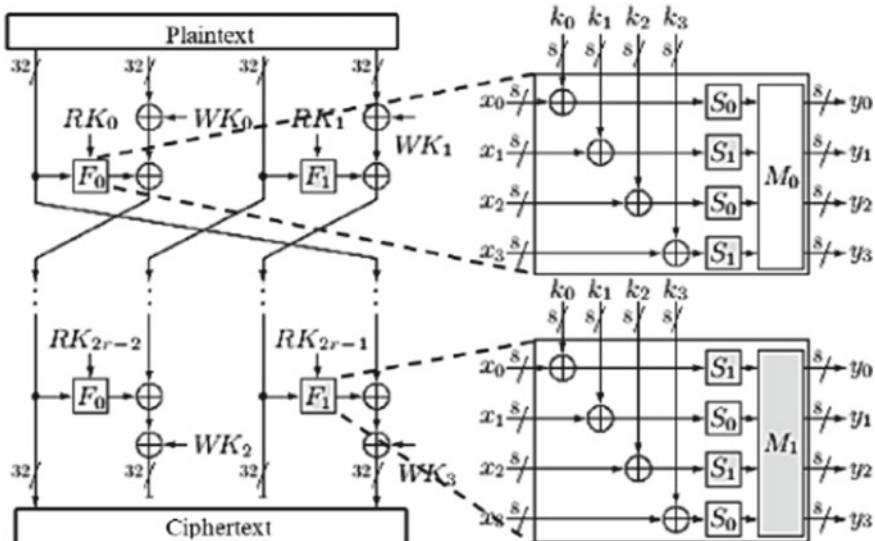
**Fig. 2** Diagrammatic representation of the Twine algorithm

Twine uses a permutation box and is believed to provide better diffusion as shown in Fig. 2.

- A cipher text is produced by processing of data through 36 rounds. As research suggests, an attack on Twine which is optimized and differential one is the most powerful which breaks Twine-128, 23-round, and 24 round. Twine-80 sufficient buffer is provided by using 36 round one.

#### 4.2 Algorithm for Round Keys (Fig. 3)

1. A randomly generated 128 bit is taken and divided into 32 blocks of 4 bits each (WK\_128)
2. RK is a  $36 \times 32$  matrix.



**Fig. 3** Round keys diagram

3. For  $i$  in 1 to 36:
4. 32 specified bits are assigned to  $RK[i]$  where  $i$  corresponds to round number
5.  $WK_{128}$  is updated by performing XOR operations with specified S-box mappings and elements from the “constants” matrix
6. Cyclic shift of  $WK_{128}$  is performed
7. Return  $RK$  (containing 36 32-bit round keys)

### **4.3 Algorithm for Encryption**

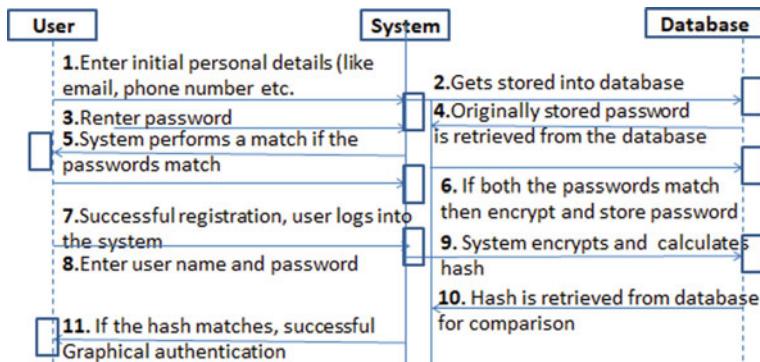
1. The function takes in 2 arguments: plain text and the round keys matrix
2. 64-bit plain text block is divided into 16 4 bit sub-blocks and stored in  $X_{16}$
3. For  $i$  in 1 to 36:
4.  $X_{16}$  is split into 2 parts - left consisting of sub-blocks 0 to 7 and right consisting of sub-blocks 8 to 16
5. The right side is updated by performing XOR operations between the S-box mapping, round key, and present expression
6. The left side is updated by using the permutation box
7. Bits from  $X_{16}$  are added to the cipher text after performing right shift and binary operation with pre-specified literals
8. Return the cipher text.

### **4.4 Graphical Encryption**

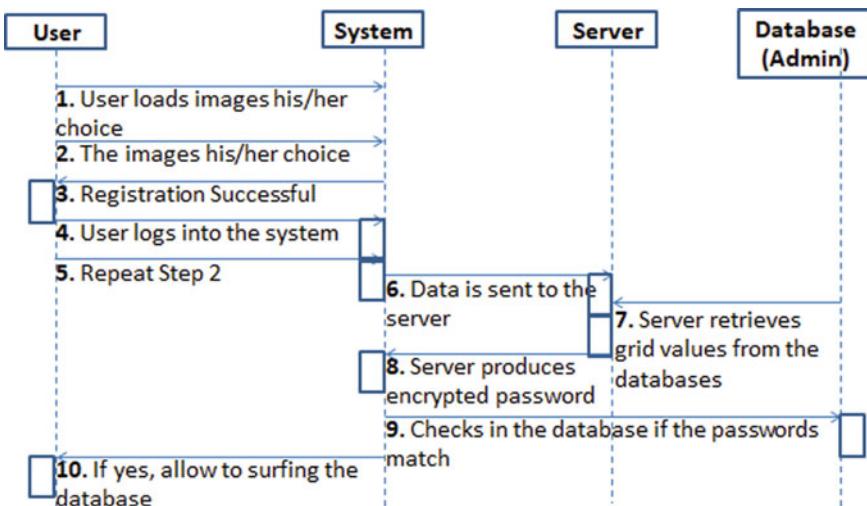
In our proposed algorithm, the image is divided into blocks, and the genetic algorithm is used to generate a key sequence of blocks that minimizes the fitness function. The fitness of a key will be calculated based on its SSIM between the encrypted and the original image. The keys with the most fitness (least SSIM) will be chosen as parents for the next generation, and the ones with least fitness (most SSIM) will be eliminated. Structural similarity index (SSIM) is a method that calculates the similarity between two images based on 3 key features, luminance contrast, and structure. The method returns a value between 0 and 1. A value of 1 means the two given images which are very similar and 0 means that the images are different to each other. The sequence diagram for text-based and graphical encryption is shown in Figs. 4 and 5.

## **5 Results and Discussion**

A comparative memory analysis when applied onto all of these helped us infer that Twine was the most suitable option due to its ability of encrypting the same size of text file without having a large amount of memory constraints. Figure 6 provides a time analysis of text encryption.



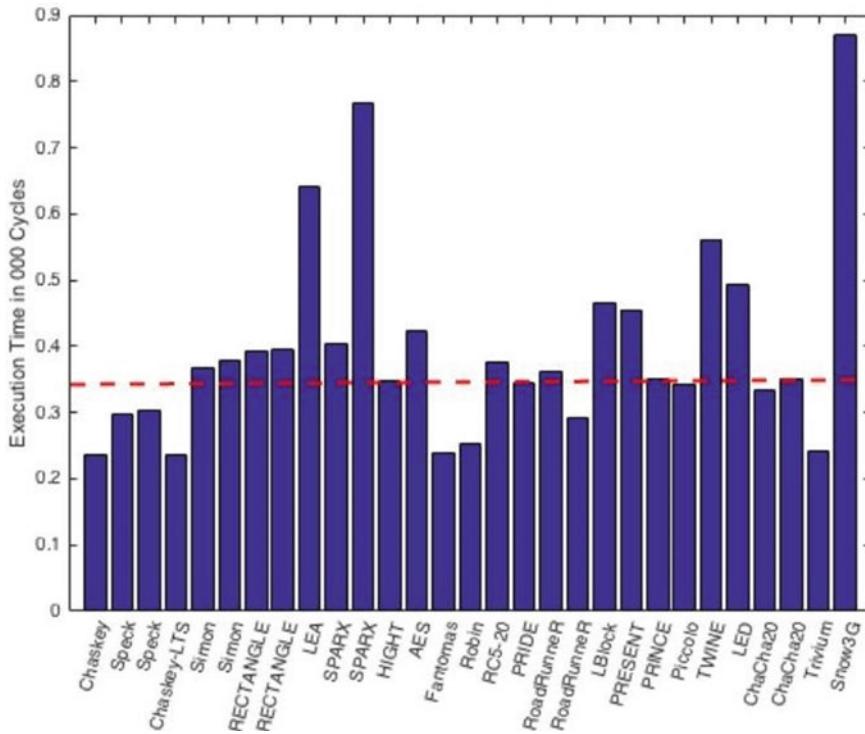
**Fig. 4** Sequence diagram for text-based encryption



**Fig. 5** Sequence diagram for graphical-based encryption

Twine proves to be one of the fastest encrypting in terms of performance as compared to other symmetric and asymmetric cryptographic algorithms. Table 1 contains the memory analysis results of text encryption in bytes.

SSIM is the structural similarity index; a way to predict the visual quality of digital television and cinema images, as well as other types of digital images and videos. SSIM is a vision-based model that considers image degradation as a subtle change in architectural knowledge while incorporating key visual events, which include both light concealment and contrast concealment terms. The difference with other methods such as MSE or PSNR is that these methods measure absolute errors. Structural knowledge is the idea that pixels have a strong tendency especially when they are close to the surface. This dependence contains important information about



**Fig. 6** Time analysis of text encryption

**Table 1** Memory analysis of text encryption in bytes

Text file size (KB)	DES	Triple DES	AES	Blowfish	Twine (128-bit)	ASCON (128-bit)
1	6899	8297	5162	4000	41.746	11.349
10	15,631	20,498	34,045	28,577	112.153	80.053
100	46,410	805,563	254,758	161,661	579.577	909.124

the structure of objects in visual space. The results are generated based on structural similarity in encrypted images and original images using genetic algorithms. Since we seek maximum potential to protect users' data, the SSIM index should be curving to a low path. Thereby, the results derived show a low percentage of similarity successfully stating the algorithm to be perfectly working. In Table 2, we capture the SSIM evaluation of graphical encryption.

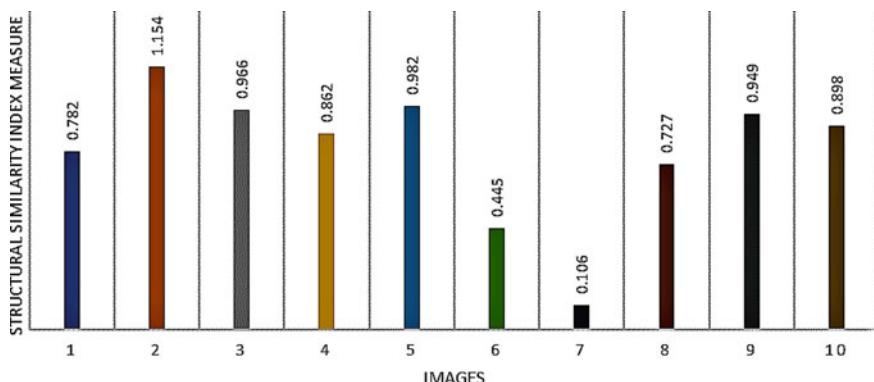
The visual chart explains the details of the test data used to compare encrypted pixel set and original image pixel set. The y-axis is the percentage, whereas the x-axis is the index of the image.

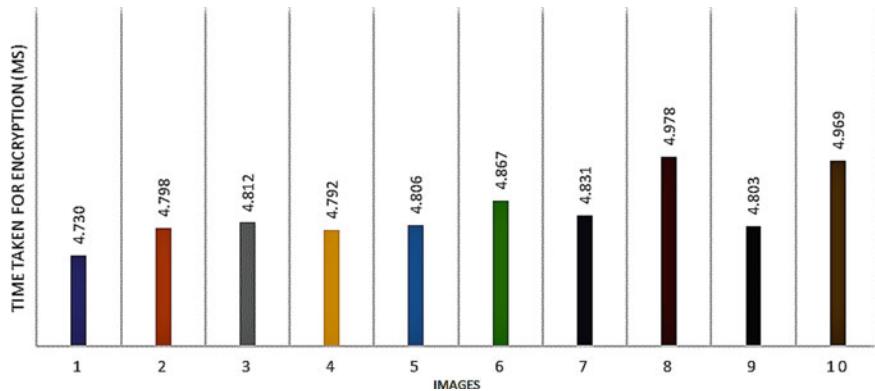
**Table 2** SSIM evaluation of graphical encryption

Filename	Encryption time	% of ssim1	SSIM1
test001.jpg	4.73	0.782	0.008
test002.jpg	4.798	1.154	0.012
test003.jpg	4.812	0.966	0.01
test004.jpg	4.792	0.862	0.009
test005.jpg	4.806	0.982	0.01
test006.jpg	4.867	0.445	0.004
test007.jpg	4.831	0.106	0.001
test008.jpg	4.978	0.727	0.007
test009.jpg	4.803	0.949	0.009
test010.jpg	4.969	0.898	0.009
<b>Average</b>	<b>4.839</b>	<b>0.787</b>	
<b>Max</b>	<b>4.978</b>	<b>1.154</b>	
<b>Min</b>	<b>4.73</b>	<b>0.106</b>	

Time taken for each image is displayed here in milliseconds. This tells us that users can have quick functioning of tasks, while image encryption safeguarding their secret passwords. Figure 7 provides the SSIM graph on evaluation of graphical encryption, and Fig. 8 provides time-based graph on evaluation of graphical encryption.

The provisional parameters for generating security attacks that are possible on each encryption methodology are tabulated to cross-check performance of each. The complex cryptographic algorithms with a greater level of security prove the point of safeguarding user's information. In both key search space and attack time, the genetic cipher proves to be much stronger than AES or DES (conventional). The key search space is vast, and attack time is also on increased levels. In Table 3, the time comparison of genetic cipher with DES and AES is accumulated.

**Fig. 7** SSIM graph on evaluation of graphical encryption



**Fig. 8** Time-based graph on evaluation of graphical encryption

**Table 3** Time comparison of genetic cipher

	DES	AES	Genetic cipher
Encryption time	068,907 mm	084,440 mm	27,069 mm
Key search space size	$4.85 \times 10^{28}$ keys	$2.31 \times 10^{57}$ keys	$1.11 \times 10^{120}$ keys
Attack time (1000 k/s)	15.41 thousand trillion days	7.34 hundred million trillion days	3.53 hundred billion trillion days

## 6 Conclusions and Future Scope

The authentication system in this paper combines graphical and text-based passwords, with multifactor authentication in a friendly intuitive system. The Twine encryption algorithm is used instead of conventional encryption ciphers. Inculcating the genetic algorithm makes it more robust and almost impossible to break. This safeguards the sensitive data from malware injection attacks, side channel attacks, authentication attacks, man-in-the-middle cryptographic attacks, network sniffing, etc.

As a future work, it can be extended to the generation of a distributed 3D password incorporating biometric and barcode authentication in addition to textual and graphical passwords. The encryption key here can be divided into three sub-keys and stored on three different authentication servers. Based on the level of security and infrastructure availability, the end user can select between 1 and 3 different levels and authentication methods. Furthermore, the scope of authentication can be extended by using graphical passwords from pre-decided images to manually upload ones, which has the chance of increasing the complexity for the encryption.

## References

1. Bhandari A, Tripathy BK, Jawad K, Bhatia S, Rahmani MKI, Mash A (2022) Cancer detection and prediction using genetic algorithms. *Comput Intell Neurosci* 2022:18. <https://doi.org/10.1155/2022/1871841>
2. Sasikumar G, Tripathy BK (2010) Innovative applications of digital image processing. In: Proceedings of National conference on recent advances in neural network (NCRANN 2010), pp A-6
3. Tripathy BK, Sooraj TR, Mohanty RK (2019) Rough set and soft set models in image processing. In: Bhattacharyya S, Pan I, Das A, Gupta S (eds) Intelligent multimedia data analysis. De Gruyter, Berlin, Boston, pp 123–144
4. Narayanan B (2019) Secure authentication using dynamic grid pair technique and image authentication. Doctoral dissertation, Dublin, National College of Ireland
5. Tripathy BK, Chandramoulli PVSSR (2016) Computer graphics and applications. Ane Books Private Limited, Chennai. ISBN: 9789385462047
6. Katsini C, Raptis GE, Fidas C, Avouris N (2018) Does image grid visualization affect password strength and creation time in graphical authentication? In: Proceedings of the 2018 International conference on advanced visual interfaces, pp 1–5
7. Parmar H, Nainan N, Thaseen S (2012) Generation of secure one-time password based on image authentication. *J Comput Sci Inf Technol* 7:195–206
8. Takada T, Yoshida M (2021) Pict-place authentication: recognition-based graphical password using image layout for better balance of security and operation time. In: CHItaly 2021: 14th Biannual conference of the Italian SIGCHI Chapter, pp 1–7
9. Calanda FB, Sison AM, Molato MRD, Medina RP (2019) Simple and secured password hiding technique for image-based authentication using a least significant bit based embedding scheme. In: 2019 IEEE 4th International conference on technology, informatics, management, engineering & environment (TIME-E). IEEE, pp 32–36
10. Mohanty M, Yaqub W (2020) Towards seamless authentication for Zoom-based online teaching and meeting. *arXiv preprint arXiv:2005.10553*
11. Ponmani E, Indhuja S, Puvilarasi R, Saravanan P, Ananthakrishnan S (2018) An enhanced least significant bit steganography to improve the effectiveness of graphical password authentication. *Int J Pure Appl Math* 119(12):13325–13335
12. Zabidi NS, Norowi NM, Rahmat RWO (2018) A usability evaluation of image and emojis in graphical password. *Int J Eng Technol* 7(4.31):400–407
13. Hamid S, Bawany NZ, Khan S (2019) AcSIS: authentication system based on image splicing. *Eng Technol Appl Sci Res* 9(5):4808–4812
14. Ghori F, Abbasi K (2013) Secure user authentication using graphical passwords. *J Independent Stud Res* 11(2):34
15. Mohamad Z, Thong LY, Zakaria AH, Awang WSW (2018) Image based authentication using zero-knowledge protocol. In: 2018 4th ICCTA. IEEE, pp 202–210
16. Khankari NB, Kale GV (2020) One time password generation for multifactor authentication using graphical password. *Int J Eng Res Gen Sci* 3(5):489–494
17. Katsini C, Fidas C, Raptis GE, Belk M, Samaras G, Avouris N (2018) Influences of human cognition and visual behavior on password strength during picture password composition. In: Proceedings of the 2018 CHI conference on human factors in computing systems, pp 1–14
18. Bianchi A, Oakley I, Kim H (2019) PassBYOP: bring your own picture for securing graphical passwords. *IEEE Trans Human-Machine Syst* 46(3):380–389
19. Mackie I, Yildirim M (2018) A novel hybrid password authentication scheme based on text and image. In: IFIP Annual conference on data and applications security and privacy. Springer, Cham, pp 182–197
20. Kadhum RN, Ali NHM (2022) Using steganography techniques for implicit authentication to enhance sensitive data hiding. *Int J Nonlinear Anal Appl* 13(1):4001–4011
21. Alia MA, Hnaif AA, Abdalla AM, Maria EMA (2018) An improved authentication scheme based on graphical passwords. *ICIC Express Lett* 12(8):775–783

22. Mishra GS, Mishra PK, Nand P, Astya R (2020) User authentication: a three level password authentication mechanism. *J Phys Conf Ser* 1712(1):012005
23. Irfan K, Anas A, Malik S, Amir S (2018) Text based graphical password system to obscure shoulder surfing. In: 2018 IBCAST. IEEE, pp 422–426
24. Lu D, Huang D, Deng Y, Alshamrani A (2018) Multi factor user authentication with in-air-handwriting and hand geometry. In: 2018 International conference on biometrics (ICB). IEEE, pp 255–262

# A Decision-Based Image Merging Technique for Server-Side Redundancy Reduction



Richa Kumari Kora and Tamal Pal

**Abstract** In a client-server architecture, plenty of images are uploaded to the server, of which many are redundant. Hence, unnecessarily consuming the memory space of the server. It also results in computation overhead for any kind of information extraction. Freeing up those spaces would result in memory availability for more images. The images could be similar to each other in terms of viewpoint etc. Merging them efficiently to produce an equivalent image reduces the quantity to a certain extent. Although few existing works combine images, they are limited to some conditions. So, we primarily design an algorithm that decides whether to merge images or not. Later, a suitable merging method is applied to merge those images. Performance of the proposed scheme is compared with the existing works. The results show that the proposed scheme is superior in terms of running time than most state-of-the-art techniques, keeping the finest quality of merging.

**Keywords** Redundancy reduction · Image registration · Image fusion · SUMO · NS3 · BRISQUE

## 1 Introduction

In a client-server architecture, the client (i.e., mobile, computer etc.) sends data to the server computer for some operation. In addition to scalar data, client devices share huge amount of multimedia data like images [1, 2], captured by the client. When such images are shared constantly among the clients, it results huge number of redundant images accumulated at the server side. As image data is more complex to process than scalar data, these redundant images not only consume server space but also add computation overhead for any type of information extraction.

One possible solution to this problem is to merge similar images in order to minimize redundancy. There are some existing approaches that deal with combining

---

R. K. Kora (✉) · T. Pal

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India  
e-mail: [richakora13@gmail.com](mailto:richakora13@gmail.com)

multiple images. Image fusion [3] is done by combining all the information together present in source images into a single one. This single image containing all important information, is more informative and has a better visualization than any single source image. Image registration [4] is done to transform different data sets of the source images into one coordinate system. It geometrically aligns the images.

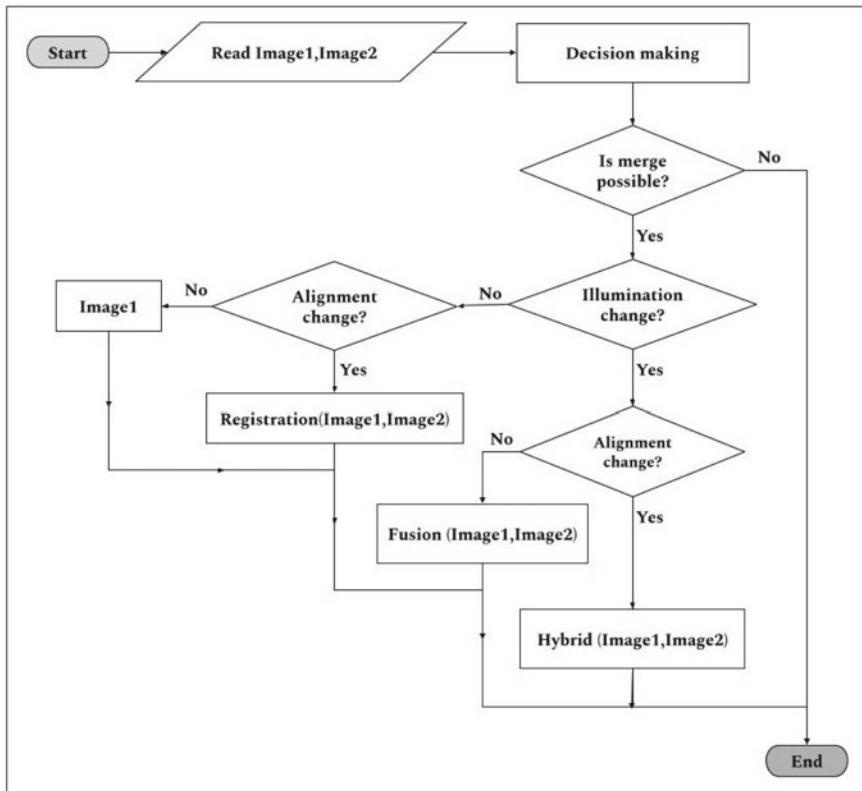
The works in [4, 5] use image registration method based on image shape features such as corners and edges, respectively. On the other hand, [6, 7] use image fusion method based on hybrid model of Hue Intensity Saturation—Discrete Wavelet Transformation (HIS-DWT) and linear-DWT technique, respectively. In both the works, the two fusion techniques are applied serially, one after the other. Moreover, a work [8] applies joint image registration-fusion method based on Scale Invariant Feature Transform—Synthetic Variable Ratio (SIFT-SVR) technique. In this, SIFT is applied firstly, followed by SVR.

Since registration and fusion methods have their respective criteria to be met, applying these solutions blindly, does not always provide desired result. Prior to merging, it is confusing whether to apply registration or fusion or both on images at the server side. Hence considering scenario, these types of solutions are constantly dependent on manual input for such decision making. On the other hand, hybrid (combined registration and fusion) approaches can be applied without manual input but these are associated with unnecessary computations as for some situations both image registration and image fusion are not required.

To overcome the said limitations, in this work, we propose a decision-based image redundancy reduction method which lessens the amount of image data by opting the suitable merging technique from the input images. As a result, this approach triggers appropriate operation (i.e., registration, fusion, both or none) depending on the situation with overall minimum computation overhead and satisfiable image quality. The paper is organized as follows. The scheme with proposed decision making algorithm is described in Sect. 2. In Sect. 3, performance of our scheme is evaluated primarily based on simulation. Finally, in Sect. 4, the paper concludes with some mention of the future scope of the present work.

## 2 Proposed Work

In this section, we propose a decision making algorithm that determines the suitable image redundancy reduction technique to be followed from given input images. We propose a decision making algorithm which decides whether two images are mergeable or not. After decision is made, the number of images is reduced by applying fusion or registration methods or both on those input images. Figure 1 depicts the flow of the proposed scheme where Image1 and Image2 are the two input images. Out of many prerequisite situations for merging pair of images, we consider the proposed decision making technique only for four different situations. For each situation, illumination, and alignment (through horizontal translation of objects) in images may or may not change as follows:



**Fig. 1** Flowchart of the scheme with proposed decision making scheme

Case 1: No illumination change and no alignment change.

Case 2: No illumination change and alignment change.

Case 3: Illumination change and no alignment change.

Case 4: Illumination change and alignment change.

In Fig. 1, prior to merging, instead of taking the decision manually, a decision making technique from input images finds the best possible approach to merge these images. This decision making technique is presented below in form of an algorithm. It takes a pair of input images and provides decision for merging technique to be applied depending on alignment and illumination change in these images, in order to get output image. Here merging techniques such as registration, fusion, and hybrid (combined registration and fusion) are considered. Decision also indicates the scenario when any type of merging is not possible. The possibility of merging is decided in terms of number of co-located edge pixels of both the input images.

In decision making algorithm, some notations and functions are used. For example,  $\text{Decision}[]$  is a vector, made of three binary components  $M, I, A$  which represent whether merge operation being possible, change in illumination and change in alignments between two input images, respectively.  $\text{Edge-detection}()$  represents a

standard edge detection technique which is used to detect edge pixels in an image. Part-of( $I, N$ ) denotes image  $I$  is vertically divided into two parts with ratio  $N : (1 - N)$  where,  $0 \leq N \leq 1$  and the first part of it is selected. For example, if  $N = 0.5$  then image is vertically divided into two equal halves and out of which the first half is selected. The function count() counts number of edge pixels in an image after edge-detection() is applied. Match() function compares number of edge pixels present in input images at corresponding locations.

---

**Algorithm 1** Algorithm for proposed decision making scheme

---

**Input:**  $Image1, Image2$

**Output:**  $Decision[]$

**Begin**

```

1:  $Decision[] \leftarrow \{Yes, Yes, Yes\}$ 
2:  $I_1 \leftarrow edge\_detection(Image1)$ 
3:  $I_2 \leftarrow edge\_detection(Image2)$ 
4:  $I_{1N} \leftarrow edge\_detection(Part\_of(I_1, N))$ 
5:  $I_{2N} \leftarrow edge\_detection(Part\_of(I_2, N))$ 
6:  $c_1 \leftarrow count(I_1)$ 
7:  $c_2 \leftarrow count(I_2)$ 
8:  $c_3 \leftarrow Match(I_1, I_2)$ 
9:  $c_4 \leftarrow count(I_{1N})$ 
10:  $c_5 \leftarrow count(I_{2N})$ 
11:  $d \leftarrow \frac{c_3}{MIN(c_1, c_2)}$ 
12: if  $((d \leq T_M) \text{ AND } (c_4 \neq c_5))$  then  $Decision.M \leftarrow No$             $\triangleright T_M : \text{A threshold}$ 
13: else if  $((c_1 == c_2) \text{ AND } (c_1 == c_3))$  then  $Decision.A \leftarrow No, Decision.I \leftarrow No$ 
14: else if  $(c_2/c_3 \approx 1)$  then  $Decision.A \leftarrow No$ 
15: else if  $(c_4 == c_5)$  then  $Decision.I \leftarrow No$ 
16: else  $Decision.A \leftarrow Yes, Decision.I \leftarrow Yes$ 
17: end if
18: return  $Decision[]$ 

```

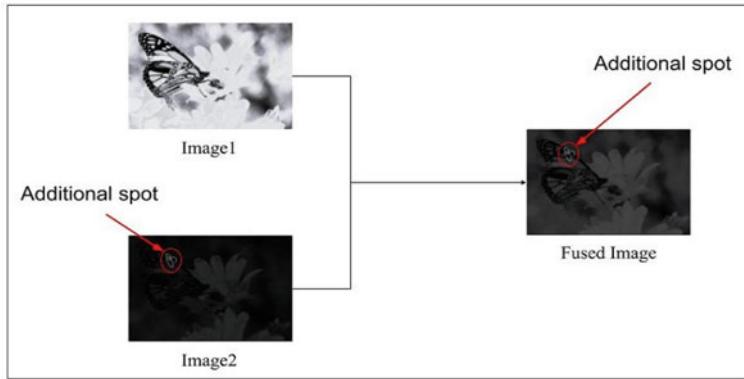
**End**

---

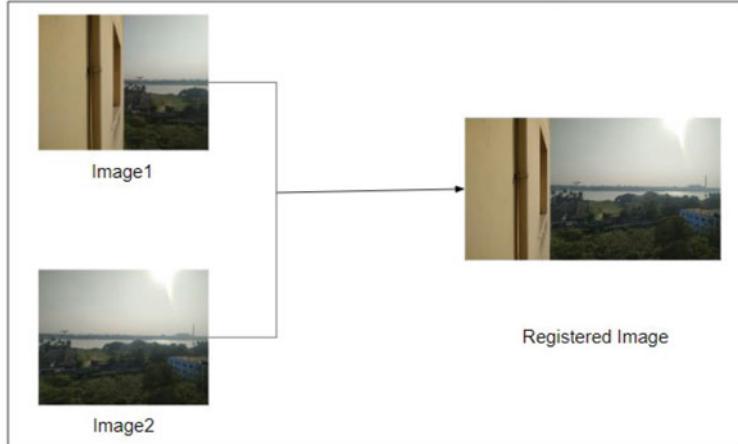
The algorithm takes any two images as input, compare them and decides for the mergeability of the two images. If the merge is not possible, the algorithm does not apply any merging technique on input images. Otherwise, an appropriate merging technique is applied in order to produce a single equivalent image from the two input images.

For this, two standard merging methods are described with the help of Fig. 2. Figure 2a illustrates merging of two images by using image fusion with two source images with different illumination and their corresponding fused image. The fused image contains all the information which is present in any of the input images and it also has a better visual quality than any of the input images taken individually.

Here, we can see that an additional information is present in Image2 which is absent in Image1. Hence, the fused image contains the additional spot and has better visibility than any single source image. Figure 2b illustrates merging of two images by using image registration where Image1 contains half of the portion of Image2 (i.e.,  $N = 1/2$ ) and vice-versa. Here, the first  $N$ th part of Image1 is horizontally concatenated [9] with Image2, which results as the registered image.



(a) Image Fusion



(b) Image Registration

**Fig. 2** Example of image fusion and registration techniques

### 3 Performance Analysis

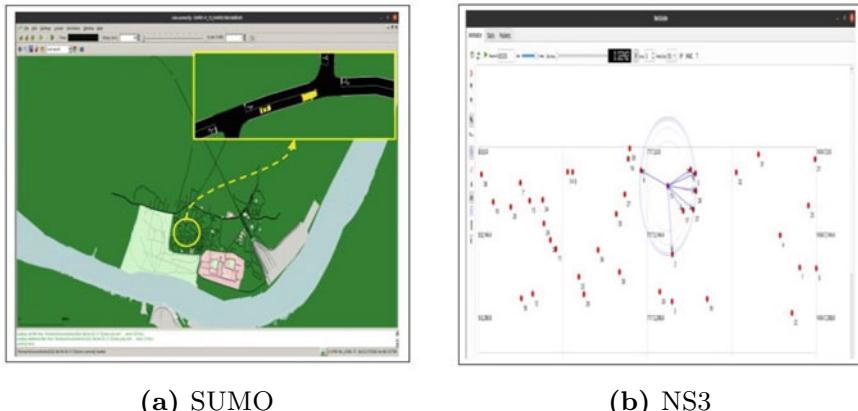
In this section, the performance of the proposed scheme, along with some existing works, is evaluated through simulation. For this simulation, we consider communication of  $400 \times 300$  color images. To detect the edge of images, we use the Canny Edge Detection method [10] as the method produces less false positives [10] as compared to other methods. We consider the value of  $T_M$  [11] as 10% of number of edge pixels. We use the linear image fusion [7] to fuse two images with less computational complexity. We simulate the merging methods of server side images with proposed decision making algorithm by considering a dataset [12] containing 100 images. The

simulation is performed on the images obtained from online resources and few are modified artificially. The existing competitor works for performance evaluation are mentioned as follows:

- TIFT (A Two-Level Hybrid Image Fusion Technique for Color Image Contrast Enhancement) [7]
- IRCMF (An image registration method based on the combination of multiple image features) [5]
- JIRF (Joint Image Registration and Fusion for Panchromatic and Multispectral Images) [8]

### 3.1 Simulation Environment

For simulation, we consider the images being captured by camera equipped vehicles over vehicular ad-hoc network (VANET) [13], transmitted to the server. Simulation is carried out by combining two simulators Simulation of urban mobility (SUMO) and Network Simulator (NS3) [14] which are shown in Fig. 3. Figure 3a shows the scenario generation in SUMO whereas Fig. 3b shows packet exchange between nodes, i.e., vehicles. However, the proposed scheme is not built for the challenges of the VANET environment. The proposed scheme is simulated on 100 sample images of a real world scene, and 50 independent runs have been done to merge the images.



**Fig. 3** Screenshots of SUMO and NS3

<b>Input Images</b>		<b>Output Images</b>			
		TIFT	IRCMF	JIRF	Proposed Method
Case-1	Image1 				
Case-2	Image1 				
Case-3	Image1 				
Case-4	Image1 				

**Fig. 4** Outputs of existing and proposed method

### 3.2 Simulation Metric

The performance of proposed as well as competitor works are evaluated based on running time and quality of output image (BRISQUE) which are defined as follows:

**Running time:** The running time [15] of an algorithm is calculated as total time taken (in seconds) by the code to execute.

**BRISQUE:** Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [16] is used for no reference image quality assessment (IQA). Due to the absence of any reference image, we have used no reference image quality metric, BRISQUE. It predicts the quality score from (0 to 100), by using trained model as described in [16]. Lesser the score, better is the quality.

### 3.3 Results and Discussion

The proposed algorithm of merging images is done on 4 cases of images as mentioned in Sect. 2. The qualities of output images of all the schemes are compared to prove the effectiveness of our scheme. A pair of input images of each case and their output is shown in Fig. 4. The pair of values indicates running time of each scheme followed

**Table 1** Comprehensive comparison of image merging of the proposed and existing schemes

	TIFT [7]	IRCMF [5]	JIRF [8]	Proposed
Case-1	55,19	29,21	43,14	28,20
Case-2	64,33	27,37	51,36	25,35
Case-3	50,54	41,85	41,30	36,28
Case-4	46,58	29,82	31,48	58,32
Avg	53,41	31,56	41,32	33,28

by BRISQUE score of the output image. We observe that overall running time of the proposed scheme is more than 2% less than the existing works. It is found to adopt the proposed decision making algorithm for better merging with a good quality of image. For example, Case-2 comprises of two input images with different illumination of the same scene without any change in alignments. We observe that the running times of the existing works [5, 7, 8], i.e., 0.94, 0.45, and 0.81, are higher than our scheme which is 0.44. Due to serial execution of two fusion methods and registration followed by fusion, the running times in TIFT and in JIRF is approx 53 and 45% more than our scheme.

We find in Case-1, our scheme has more than 50% less running time than TIFT and JIRF due to the proposed decision making algorithm in our scheme. On the other hand, JIRF has the least BRISQUE score, hence the better quality due to serially execution of registration method followed by fusion method. In Case-3 and Case-4, we observe that our scheme produces the better quality of image than the existing works as none of the existing works produces the desired output in both the cases. The running time of our scheme is higher than existing schemes in Case-4 due to successive execution of the proposed decision making algorithm followed by the registration method and further followed by fusion method.

Table 1 presents a comparison of our scheme with some of the existing works (TIFT [7], IRCMF [5] and JIRF [8]) and shows the average result of 100 images. The table provides comprehensive comparison including simulation results. The pair of values indicates total running time followed by average BRISQUE score for the entire set input images available at the server side. On average, the proposed scheme produces more than 14% better quality of images than the competitor works. Also, the proposed scheme takes more than 13% less running time than most of the existing works. For example, in Case-2 and Case-3, we observe that the proposed scheme dominates the existing works in terms of running time as well as BRISQUE score. We observe in Case-1 that TIFT and JIRF produce better image quality than our scheme due to two merging methods in each. Also, we observe in Case-4, the proposed scheme has the highest running time due to successive decision making followed by registration further followed by fusion. However, the resultant image has the better quality than the existing works. On an average, TIFT has the least running time as direct registration is applied to merge the images. However, our scheme produces better quality of images as the BRISQUE score is least among all the works.

Practically, the existing works are capable to reduce the images at the server by 50% whereas, our scheme is capable to reduce the redundancy by 46%. It is due to the fact that existing works do not undergo any prior decision making to check if the images can be merged or not. On the other hand, the proposed method, first decides and check for similar images and then merging is done. If we consider Case-3 in Fig. 4, the TIFT method undergo image fusion even though the images are aligned differently. As a result, an undesired image is produced as output. Similarly, various cases include unnecessary merging and an irrelevant image is produced as output. But, in our case, merging is done only if two images are mergeable as decided.

## 4 Conclusion

In this work, a low-overhead redundancy reduction method is proposed for successfully reducing the volume of image data transmitted to the server. On an average, the proposed scheme takes 47 and 13% less running time over the competitor scheme, TIFT and JIRF and 6% more than IRCMF. In terms of quality of merged images, the proposed scheme is 46, 50, and 14% better than the existing works. Although competitor works provide 8% more redundancy reduction, these works often end up with undesirable outputs. The comparative results show that in most of the cases our scheme is superior not only in terms of less running time but also in getting a better resultant image out of two mergeable images. The proposed work considers only a limited number of cases for merging images. The present work may be improved by considering other complex scenarios of merging images by considering complex geometric transformations. Moreover this approach can also be extended by incorporating VANET-specific challenges in it.

## References

1. Zhang B, Xia Q, Han F (2015) Multi-point-to-point image transmission system based on TCP/IP protocol. In: 5th international conference on instrumentation and measurement. Computer, communication and control. IEEE Press, Qinhuangdao, pp 1621–1624
2. Balakhontseva H, Antoshchuk S, Sieck J (2015) A client server application for the recognition of artefacts in a Museum. In: International conference on emerging trends in networks and computer communications. IEEE Press, Windhoek, pp 105–109
3. Zhao X (2017) Image fusion based on IHS transform and principal component analysis (PCA) transform. In: International conference on computer technology. Electronics and Communication. IEEE Press, Dalian, pp 304–307
4. Sui H, Song Z, Gao D, Hua L (2017) Automatic image registration based on shape features and multi-scale image segmentation. In: 2nd international conference on multimedia and image processing, pp 118–122. IEEE Press, Wuhan
5. Wang G-k, Xu H-p, Zhang H (2016) An image registration method based on the combination of multiple image features. In: IEEE international geoscience and remote sensing symposium, pp 2803–2806. IEEE Press, Beijing

6. Cai C, Ding X (2018) Fusion of infrared and visible image based on HIS and wavelet transform. In: 30th Chinese control and decision conference. IEEE Press, Shenyang, pp 2662–2667
7. Kumar V, Aziz S, Shahnawazuddin S (2021) A two-level hybrid image fusion technique for color image contrast enhancement. In: 12th international conference on computing communication and networking technologies. IEEE Press, Kharagpur, pp 1–6
8. Zhang Q, Cao Z, Hu Z, Jia Y, Wu X (2015) Joint image registration and fusion for panchromatic and multispectral images. *IEEE Geosci Rem Sens Lett* 12(3):467–471
9. Concatenate images using OpenCV in Python. <https://www.geeksforgeeks.org/concatenate-images-using-opencv-in-python/>. Accessed 15 Mar 2023
10. Biswas R, Sil J (2012) An improved canny edge detection algorithm based on Type-2 fuzzy sets. In: 2nd international conference on computer, communication, control and information technology. Procedia Technology, vol 4, pp 820–824
11. Thresholding (image processing). [https://en.wikipedia.org/wiki/Thresholding\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing)). Accessed 15 Mar 2023
12. Image Dataset. <https://drive.google.com/drive/folders/1avL59-zo6k1hUN-M5NubSo769n8rCyjh>
13. Pang F, Bai X (2019) Traffic image acquisition and compression technology in vehicular ad hoc network. In: IEEE international conference on parallel & distributed processing with applications. Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking. IEEE Press, Xiamen, pp 250–255
14. Amina B, Mohamed E (2018) Performance evaluation of VANETs routing protocols using SUMO and NS3. In: IEEE 5th international congress on information science and technology. IEEE Press, Marrakech, Morocco, pp 525–530
15. Python—Measure time taken by program to execute, <https://www.geeksforgeeks.org/python-measure-time-taken-by-program-to-execute/>. Accessed 09 Mar 2023
16. Image Quality Assessment: BRISQUE. <https://learnopencv.com/image-quality-assessment-brisque/>. Accessed 09 Mar 2023

# Handling Class Imbalance Problem Using Support Vector Machine



Mehwish Naushin, Ankur Das, and Asit Kumar Das

**Abstract** The class imbalance problem makes it difficult to use a classification model. The model may not be trained appropriately due to the availability of a lower number of objects of a minority class. When we need real-time or critical prediction, this problem can be precarious. So the model, when needed for real-time or critical prediction, the class imbalance problem may play a significant role in the hazard regarding the training and subsequent functioning of the model. To avoid such a hazard, we need to balance the dataset. The proposed model uses a support vector machine (SVM) to determine the outliers. Those outliers deal with the class imbalance problem. Our study analyzes different classification algorithms employed to predict stroke or non-stroke classes based on health diagnosis data collected from Kaggle. After balancing the data, we conducted several experiments to test the model's functioning using various traditional classification models.

**Keywords** Support vector machine · Outliers detection · Disease identification · Data imbalance problem

## 1 Introduction

The class imbalance problem is a new age topic explored by machine learning and data mining researchers [1]. When the observation of one class is more than the other classes, one class is overwhelmed, called the majority class, while the other is called the minority class. However, in many applications, the class with lower instances is of more importance. This imbalance problem increases when the class of interest has fewer instances than the majority class; for example, consider cancer versus non-cancer or fraud versus un-fraud [2].

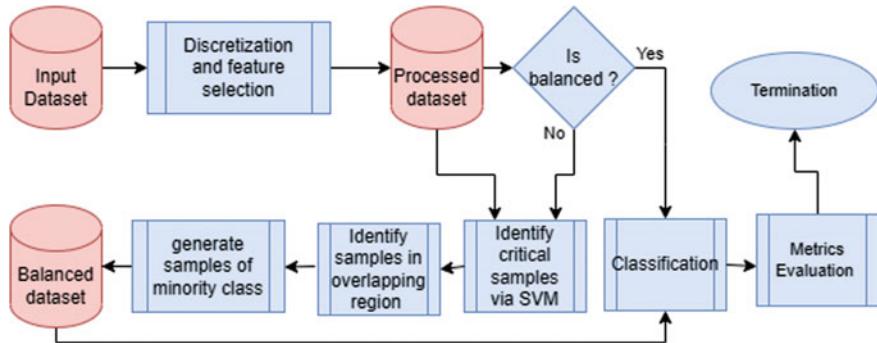
---

M. Naushin · A. K. Das (✉)

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India  
e-mail: [akdas@cs.iests.ac.in](mailto:akdas@cs.iests.ac.in)

A. Das

Department of Software Engineering, Concordia University, Montreal, Canada



**Fig. 1** Sequence of the process of the proposed work

There are different ways [3], like algorithm level, data level, or hybrid level, to address the class imbalance. We have used the data level method for balancing the dataset. This method includes various sampling approaches to treat the data. Some approaches that researchers use to deal with imbalanced datasets are:

1. **Choosing Proper Evaluation Techniques:** The classification model is generally evaluated with the help of the accuracy of the model. But it fails to assess the model properly when it applies to an imbalanced dataset. It requires measuring precision, recall, and  $F_{\text{score}}$  of the model in such a dataset.
2. **Resampling:** It includes oversampling and undersampling. In oversampling, researchers upsample the minority class using the replacement technique. This method experiences the addition of illegitimate samples, which is still a thriving research area. In undersampling, researchers remove samples from the majority class to balance them with the minority. Thus, it removes essential data and does not provide precise classification outcome.
3. **Synthetic Minority Oversampling Technique (SMOTE):** It is a statistical technique that is most popularly used for oversampling minority class objects. It does not add duplicate records but synthesizes new instances from the existing minority class instances. The major drawback of SMOTE is that it concentrates on local information of minority classes which does not generate a diverse dataset, which may cause the overfitting problem in the model.

There are also issues due to the overlapping of instances of different classes [4], which relates to the degree of separability between classes of the data. High overlapped classes add to the complexity. Thus, our objective is to balance the imbalanced dataset by overcoming the above limitations. To solve the class imbalance problem, the proposed work first employs a support vector machine to collect support vectors. The set of samples within the  $\epsilon$ -neighborhood of the support vectors is considered as the overlapping set that generates new samples of the minority class. The sequence of the process of the proposed work is demonstrated in Fig. 1.

Therefore, the main contributions in this paper are as follows:

1. The dataset is gathered from Kaggle, which may be balanced or imbalanced. Initially, we preprocess the dataset to remove irregularities in the data. Then we discretized the dataset and selected the most relevant features using the feature selection algorithm.
2. The support vector machine (SVM) is used to the preprocessed data to obtain the support vectors. Next, every instance of the dataset in the  $\epsilon$ -neighborhood of support vectors is considered an overlapping set of instances lying in the overlapping region of the sample space.
3. The instances in the overlapping set are used to generate minority class instances with the help of simple mathematical concepts. We make the dataset balanced in two ways: (i) by removing all majority class observations from the overlapping region and (ii) by not removing the majority class observations from the overlapping region. In both cases, we have generated the minority class instances as many as required to make the dataset balance.
4. Finally, the balanced dataset is applied to different classification models to measure the performances. It is observed that the balanced dataset generated by the proposed method gives better classification performance with accuracy, precision, recall, and  $F$ -measure metrics than the imbalanced dataset.

The rest of the paper is along these lines: Sect. 2 gives the background study of related work. The framed SVM-based oversampling technique is discussed in Sect. 3. The result of the experiment displaying the method's effectiveness is described in Sect. 4. In the last, this paper's conclusion and future scope are discussed in Sect. 5.

## 2 Related Work

Research continually improves the socioeconomic aspect of society. Data balancing is one of the essential factors that directly or indirectly improve the research work in pattern recognition. In some cases, imbalanced data can drastically affect a model's performance. Researchers have proposed various strategies for class imbalance fall-outs. Sharma et al. [2] came up with the finding to assess sentiment for a class imbalanced code-mixed data using Levenshtein distance. Subalalitha and Srinivasan [5] have proposed a novel hybrid approach that combines SMOTE and GAN as SMO-Tified GAN. They worked to vanquish SMOTE's and GAN's deficiencies by transfer learning notions. They extracted information about minority classes from SMOTE and then made use it to GAN. Li et al. [6] have proposed a divide-and-conquer strategy-based hybrid structure to handle the class imbalance case. In the divide step, SVM detects fraud transactions based on which overlapping subsets are formed. Moreover, in the conquer step, the ANN classifiers are applied to the overlapping subsets to improve the method's efficiency. Thanh Vo et al. [7] presented an effective structure that utilizes an oversampling strategy named as FJD-OT. This framework applies two modules to get feature vectors, preprocessing and feature extraction. The method uses SVM SMOTE to balance the training set and uses a logistic regression

model for prediction. Jang et al. [8] addressed the issue in a data-driven approach by proposing novel training architecture(ST) where they partitioned training data into mutually exclusive subsets using KL-divergence. They sequentially trained the learner and used EWC to stabilize knowledge taken from the previous partition, and finally, the CNN + BiLSTM model was used to test the proposed method. Lee and Park [9] solved the data imbalance problem using the GAN model. They compared the classification performance of a single Random Forest (RF) model, which classifies minor class data without resampling, with the classification performance of the GAN-RF model after resampling based on GAN. Banerjee et al. [10] tackled class imbalance in sarcasm detection by imposing SMOTE. They did an experimental analysis of the effect of oversampling in both small and large datasets. Shafqat et al. [11] implemented conditional Wasserstein GAN with gradient penalty to generate tabular data containing numerical and categorical values. They also classified augmented auxiliary classifier loss to enforce the model for generating instances of minority class explicitly. They have designed a discriminator architecture with PacGAN to receive m-packed samples as input instead of a single input. Yafooz and Alsaeedi [12] tried to discover the opinions of YouTube users on herbal treatment videos related to diabetes through the analysis of user comments by using MLC. They examined the impact of four representation methods on ADHTD to observe the performance of MLCs by normalization, stop word removal, tokenization, and Arabic stemming. They also applied  $n$ -gram with imbalanced dataset techniques like oversampling, undersampling, and SMOTE, where SMOTE worked better. Suh et al. [13] proposed a classification enhancement GAN to enhance the quality of generated synthetic minority data and to improve prediction accuracy. Furthermore, they proposed an ambiguity reduction method using generated synthetic minority data for multiple similar classes that degenerate the classification accuracy. Here, the objective formulation of WGAN-GP is deployed to improve stability. They modified the classifier for a generated image in CEGAN to reduce the impact of noise input and ambiguity between classes via a conditional generative model with class subsets determined by feature extraction and clustering. Imran and Yang [14] addressed the data imbalance issue for the sentiment analysis of user's opinions task on two educational feedback datasets utilizing synthetic text generation deep learning models, SentiGAN and CatGAN. BLUE, NLLgen, and NLLdiv are the metrics they used to find the quality of generated text in reference to the original text and evaluate the model's performance in sentiment analysis. Mollas et al. [15] created a dataset with two variations, a balanced binary and a multilabel one. They focused on HS scenarios and provided an analysis of this use case. Then a series of experiments was conducted to determine the performance of the dataset using SOTA techniques. In all these works, the researchers tried to generate new data to balance the training set, but they have yet to consider the outliers and overlapping regions. Keeping this in mind, we have proposed SVM-based oversampling technique using simple mathematical concepts.

### 3 Proposed Methodology

The presented work gathers the imbalance data from Kaggle and makes it balanced using an SVM-based [16, 17] sampling technique. The proposed method finds the overlapping region and removes the major class instances. Finally, based on an efficient clustering algorithm, it performs oversampling on minor class datasets. Initially, we removed the abnormalities and voids from the dataset and normalized the data before the further application of preprocessing techniques.

#### 3.1 Discretization and Feature Selection

We have discretized the dataset using the Weka tool [18] so that each continuous feature has ten discrete values. We then classified the original and discretized data with several classifiers and realized that discretized data provided similar results. That is why we have discretized the data to lower the complicity of the proposed model. All the features are not necessarily essential, and sometimes, unnecessary features may downgrade the implementation result and increase the difficulties during the model's training. Thus, feature selection is significant before building the classification model. Some feature selection techniques [19, 20] are filter, wrapper, and embedded methods. In the presented work, we have used the 'filter method.' This method is perfect for removing duplicated, correlated, and redundant features.

We have used various feature selection algorithms, like a combination of CfsSubsetEval and BestFirst, Ranker with ClassifierAttributeEval or CorrelationAttributeEval using weak tool [18], and found the combination of features that provides the best classification results. We have eight features from the health data, which include Gender, Age, Hypertension, Heart\_disease, Ever\_married, Work-type, Avg\_glucose level, and Smoking\_status, after applying these feature selection algorithms.

#### 3.2 Finding Overlapping Region

We have applied the SVM training method using SkLearn in the colab notebook and collected the SVM support vectors. Some of these support vectors are of the majority class, and others are of the minority class. For each support vector, we have considered an  $\epsilon$ -neighborhood region as five units, and all the instances of the region are collected. Thus for each support vector, we get a set of instances. We consider all the support vectors and combine all the sets of instances to get the overlapping set (OVPset) of instances, where both the instances of the major class and minor class exist. Then, the set forms an overlapping region, and other regions contain the instances of any one class, either majority class or minority class. We divide the instances in the overlapping region into two subsets, OVPmajor and OVPminor,

where OVPmajor is the set of observations of the major class and OVPminor is the set of observations of the minor class. We have done undersampling by removing all the instances in OVPmajor as they are closest to the support vectors of the same class. The oversampling is done by generating samples of minority class with the help of the instances which are not in the set OVPminor.

### 3.3 Generate Minority Class Instances

The instances in OVPminor are closed to the support vectors of the minority class, so a slight change in the boundary of SVM may misclassify them. That is why we have excluded the instances in OVPminor during the generation of instances of the minority class. Consider  $S_1$  and  $S_2$  be the set of instances of the majority class and minority class of the dataset, respectively. Therefore, after undersampling, the set  $S_1 - OVPmajor$  contains all the majority class instances. If  $S_1 - OVPmajor$  and  $S_2$  contain almost the same number of instances, then the dataset is balanced, and we need not oversample. Otherwise, we proceed further for generating minority class instances using  $S = S_2 - OVPminor$ . There are many clustering algorithms [21] [22] with their merits and demerits in the literature. We have used an agglomerative clustering algorithm [23] on dataset  $S$  for our proposed oversampling process. We have chosen this clustering algorithm for its simplicity, and also it does not require a predefined number of clusters as input of this algorithm. Let the algorithm gives  $k$  number of clusters,  $C_1, C_2, \dots, C_k$ . To make the dataset balance, we need to create almost  $N = |S_1 - OVPmajor| - |S_2|$  instances of the minority class, where  $|S_1 - S_2| \geq |OVPmajor|$ . Since different cluster contains a different number of instances, so we generate a different number of minority class instances from different clusters. Let  $q_i$  be the number of minority class instances need to create from cluster  $C_i$ , for  $i = 1, 2, \dots, k$ , where  $q_i$  is given by Eq. (1). Now we will discuss how  $q_i$  number of instances are generated from cluster  $C_i$ , for  $i = 1, 2, \dots, k$ . Let,  $R_i = (x_{i1}, x_{i2}, \dots, x_{id})$  be the representative of cluster  $C_i$ , where  $d$  is the dimension of the feature vectors. We select randomly an instance, say  $O_j = (y_{j1}, y_{j2}, \dots, y_{jd})$  from  $C_i$ . Next, we select a natural number, say  $l$  randomly from  $[1..d]$ , where  $1 \leq l \leq d$ . Then we create  $O'_j$  by considering all the feature values from  $O_j$  except the  $l$ th feature value, which is taken from  $l$ th value of  $R_i$ , i.e.,  $O'_j = (y_{j1}, y_{j2}, \dots, y_{j(l-1)}, x_{il}, y_{j(l+1)}, \dots, y_{jd})$ . We repeat  $q_i$  times this random selection of instance from  $C_i$ , and each time only one new instance is generated by modifying only one feature value by the representative. Thus, the generated instances are expected to be similar to the cluster representative. This process is repeated for all the  $k$  clusters. Thus, the generated  $\sum_{i=1}^k q_i \approx N$  instances are of the minority class, and the dataset becomes balanced.

$$q_i = \lfloor \frac{N - |S_2|}{\sum_{j=1}^k |S_j|} \rfloor \quad (1)$$

The pseudocode of the presented sampling algorithm is given in Algorithm 1.

---

**Algorithm 1:** Sampling for balancing dataset ( $DS$ )

---

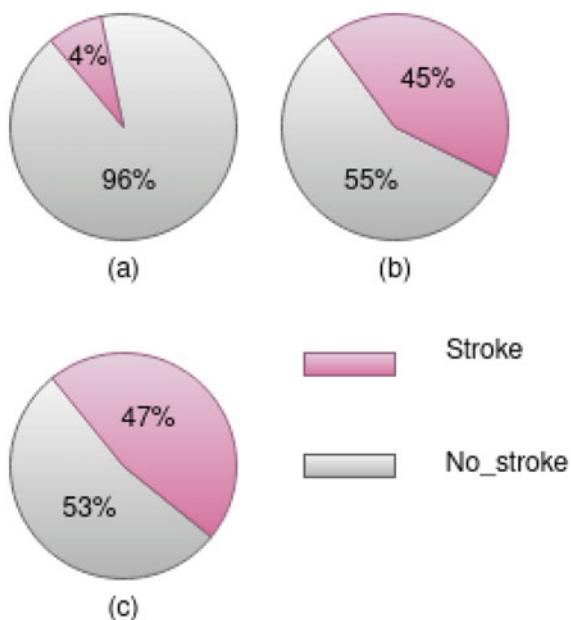
**Input:**  $DS = (U, D)$ , the unbalanced dataset

/\*  $U$  is the set of samples and  $D$  is the decision class, namely the majority and minority class \*/

**Output:**  $DS = (U, D)$ , the dataset after balancing

- 1 Let  $U = U_{major} \cup U_{minor}$ , where  $U_{major}$  and  $U_{minor}$  are the set of observations of majority category and minority category, respectively;
  - 2  $NEB_{major} = NEB_{minor} = \emptyset$ ;
  - 3 Apply SVM on  $DS$  to get support vectors
  - 4 Let  $SV_{major}$  = Set of support vectors of major class;
  - 5 Let  $SV_{minor}$  = Set of support vectors of minor class;
  - 6 **for** each  $x \in SV_{major}$  **do**
  - 7    $NEB_{major} = NEB_{major} \cup \epsilon - neighbor(x)$ ;
  - 8    $U_{major} = U_{major} - NEB_{major}$                                   /\* undersampling \*/
  - 9 **for** each  $x \in SV_{minor}$  **do**
  - 10    $NEB_{minor} = NEB_{minor} \cup \epsilon - neighbor(x)$ ;
  - 11  $S = U_{minor} - NEB_{minor}$ ;
  - 12 Apply clustering algorithm described in [23] on  $S$ ;
  - 13 Let the clusters obtained are  $C_1, C_2, \dots, C_k$ ;
  - 14 For each  $i = 1$  to  $k$ , let  $R_i$  is the representative (centroid) of  $C_i$ ;
  - 15 **for**  $i = 1$  to  $k$  **do**
  - 16   Let  $R_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ;
  - 17 **for** each  $C_i$  **do**
  - 18    $New\_C_i = \emptyset$ ; /\* newly generated instances by cluster  $C_i$  \*/
  - 19   Compute  $q_i$  using Eq. (1);                                  /\*  $q_i$  is the no of instances created \*/
  - 20   **for**  $j = 1$  to  $q_i$  **do**
  - 21     Randomly select an instance  $O_j = (y_{j1}, y_{j2}, \dots, y_{jd})$ ;
  - 22      $l = \text{rand}(1..d)$ ;
  - 23     Set  $O'_j = (y_{j1}, y_{j2}, \dots, y_{j(l-1)}, x_{il}, y_{j(l+1)}, \dots, y_{jd})$ ;
  - 24      $D(O'_j) = \text{minor class}$ ;                                  /\* Set class of new instance \*/
  - 25      $New\_C_i = New\_C_i \cup \{(O'_j, D(O'_j))\}$ ;
  - 26    $U_{minor} = U_{minor} \cup New\_C_i$ ;
  - 27  $U = U_{major} \cup U_{minor}$ ;
  - 28 **return**  $DS$ ;
-

**Fig. 2** Percentage of instances. **a** Imbalanced. **b** Balanced by oversampling. **c** Balanced by oversampling and undersampling



#### 4 Result and Discussion

We have worked with ‘Health diagnosis for Stroke,’ which is an imbalanced and binary class dataset, as shown in Fig. 2a, where 96% instances are of majority class (stroke) and only 4% instances are of minority class (nostroke). The dataset is preprocessed by removing irregularities and applying discretization and feature selection algorithms. Finally, the SVM and clustering algorithms are applied for generating new minor class (i.e., stroke) instances, and the corresponding balanced set of data are shown in Fig. 2b and c. We have applied both under sampling and oversampling techniques in the proposed work. But for experimental analysis, we have considered two different cases: (i) Fig. 2(b) is obtained when we apply only the oversampling technique, i.e., we did not remove any major class instances, and (ii) Fig. 2c is corresponding to the application of both oversampling and undersampling techniques for generating the equal number of new samples of minority category.

Thus, we have used three different types of dataset. Initially, the collected imbalance dataset (ID) with 4% positive class (i.e., stroke) instances and 96% negative class (i.e., nostroke) instances is applied to various traditional classifiers. Next, the dataset is made balanced by applying only oversampling technique and named as BO, where 45% instances are of positive class (i.e., stroke) and 55% instances are of negative class (i.e., nostroke). In this case, only the existing minority class (i.e., stroke) instances are used to generate new minority instances. Finally, the dataset BS with 47% minority category observations and 53% majority category observations is made by using both oversampling and undersampling techniques. Here, the

majority class instances in overlapping region are removed (i.e., undersampled), and the minority class instances are generated (i.e., oversampled) from the existing minority class instances using proposed SVM-based sampling technique. All three datasets (i.e., ID, BO, and BS) are applied to various traditional classifiers such as BayesNet, logistic, multilayer perceptron (MLP), SGD, SimpleLogistic, SMO, Voted Perceptron, IBk, KStar, AdaBoost, Attribute Selected Classifier (ASC), Bagging, Classification Via Regression (CVR), Filtered Classifier, Iterative Classifier Optimizer (ICO), LogitBoost, Multi-Class Classifier (MCC), MCC Updateable, Random Committee, Randomizable Filtered Classifier (RFC), Random Sub-Space, Decision Table, JRip, PART, Hoeffding Tree, J48, LMT, Random Forest, Random Tree, REP-Tree, to validate the presented method.

The performance of the classification algorithms based on the metrics such as accuracy ( $A$ ), precision ( $P$ ), recall ( $R$ ), and  $F$ -measures ( $F$ ) (computed using Eqs. (2)–(5), respectively where the terms in the equations have their usual meaning) is listed in Table 1. In Table 1, it is observed that for dataset ID, almost 95% accuracy is achieved in most of the classification models, but when we consider the precision, the models provide undefined values in most of the cases which is denoted by hyphen (-). This occurs because there aren't many samples from the positive class in the dataset, and the model misclassifies these while correctly classifying the samples from the negative class. So, when 0 is divided by 0, its yield is undefined. Similar scenario happens for recall, and as a result, the  $F$ \_measure is also undefined. So, even though the accuracy is quite good, the models are not reliable. So, there is a pressing call to handle this issue. This is why we have balanced the data and then applied the same classifiers to check for the value we have obtained, if they have improved, for all the metrics we used earlier. In Table 1, for dataset BO, it is observed that the classifiers give a satisfying accuracy ranges from around 82% to 96% and balanced precision, recall, and  $F$ -measure values. Thus, though the accuracy is decreased, the values of the other metrics are increased which implies that the classifiers could predict the minority instances more accurately. Similarly, for BS dataset, generally we get increased accuracy, which scales from 83% to 97%. However, though the classifier VotedPerceptron has reduced the accuracy to 92% from 93%, the rest of the models either have increased or the same accuracy that is achieved for BU dataset. So in case of dataset BS, we achieve satisfactory performance in terms of accuracy as well as other metrics of the classifiers, which implies that undersampling and oversampling together perform better prediction of both majority category and minority category observations.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (2)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

**Table 1** Performance assessment with several classifiers and datasets

Classifier	ID (in %)				BO (in %)				BS (in %)			
	A	P	R	F	A	P	R	F	A	P	R	F
BayesNet	94	21	19	20	82	99	63	77	83	98	63	76
Logistic	95	—	0	—	94	95	91	93	95	96	91	94
MLP	94	16	04	07	96	98	95	93	97	98	94	96
SGD	95	—	0	—	95	96	91	93	95	97	91	94
SimpleLogistic	95	—	0	—	94	97	90	93	95	98	90	94
SMO	95	—	0	—	95	96	91	93	95	97	91	94
Voted perceptron	95	—	0	—	93	91	91	91	92	91	91	91
IBk	92	14	13	14	95	94	94	94	96	96	94	95
KStar	95	15	01	03	96	97	93	95	97	99	93	96
AdaBoost	95	—	0	—	95	98	91	94	96	98	92	96
ASC	95	—	0	—	96	98	92	95	96	98	93	96
Bagging	95	—	0	—	96	98	93	96	97	98	94	96
CVR	95	—	0	—	96	98	93	96	97	98	94	96
FilteredClassifier	95	—	0	—	96	98	93	95	97	98	94	96
ICO	95	—	0	—	96	98	93	95	97	98	94	96
LogitBoost	95	—	0	—	96	98	93	95	97	98	94	96
MCC	95	—	0	—	94	95	91	93	95	96	91	94
MCC updateable	95	—	0	—	95	96	91	94	95	97	91	94
Random committee	93	11	07	09	95	95	94	94	96	96	94	95
RFC	93	07	05	06	95	94	94	94	96	96	94	95
RandomSubSpace	95	—	0	—	96	98	91	95	96	98	92	96
Decision table	95	—	0	—	96	98	93	93	97	98	94	96
JRip	93	23	02	03	96	98	93	96	97	98	93	96
PART	95	18	03	06	96	97	93	95	97	98	98	97
Hoeffding tree	95	—	0	—	95	98	90	94	95	97	91	94
J48	95	—	0	—	96	98	93	96	97	98	94	96
LMT	95	—	0	—	96	98	93	95	97	98	94	96
Random forest	94	10	03	05	92	89	95	92	97	98	94	96
Random tree	93	15	13	14	96	97	93	95	97	98	94	96
REPTree	95	14	01	01	96	97	93	95	97	98	94	96

$$F\text{-measure} = \frac{2.\text{Precision}.\text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

## 5 Conclusion

The imbalanced dataset creates problems in the prediction of minority class samples, which becomes acute, especially in the case of disease identification. To tackle this problem and improve classification quality, we have applied improved resampling

methods using SVM. The proposed approach balances the data effectively and provides more instances for the minority category, enhancing the performance of the classification models. We have generated the overlapping region where the objects in the region are denoted as critical, as a slight change in their positions may misclassify them. However, this overlapping region is determined by the parameter  $\epsilon$ , which is experimentally set. In future work, we will apply the concepts of rough set theory to generate this overlapping region for which no such parameter is required. The SVM is time-consuming and cumbersome for a multiclass problem, where rough set theory is comparatively easier. The proposed work is evaluated with the help of different classifiers but is not compared with similar types of state-of-the-art procedures, which is kept as the future work for our presented method.

## References

1. Thabtab F, Hammoud S, Kamalov F, Gonsalves A (2020) Data imbalance in classification: experimental evaluation. *Inf Sci* 513:429–441
2. Sharma A, Singh PK, Chandra R (2022) SMOTified-GAN for class imbalanced pattern classification problems. *IEEE Access* 10:30655–30665
3. Ali H, Salleh MNM, Saedudin R, Hussain K, Mushtaq MF (2019) Imbalance class problems in data mining: a review. *Indonesian J Electr Eng Comput Sci* 14(3):1560–1571
4. Vuttipittayamongkol P, Elyan E, Petrovski A (2021) On the class overlap problem in imbalanced data classification. *Knowl Based Syst* 212:106631
5. Subalalitha CN, Srinivasan R (2021) Sentimental analysis from imbalanced code-mixed data using machine learning approaches. In: Distributed and parallel databases, pp 1573–1578
6. Li Z, Liu G, Jiang C, Huang M (2021) A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection. *Expert Syst Appl* 175(114750). ISSN 0957-4174
7. Thanh Vo M, Nguyen T, Sharma R, Vo AH, Le T (2021) Dealing with the class imbalance problem in the detection of fake job descriptions. *Comput Mater Continua* 68(1):521–535
8. Jang KCSSJ, Kim Y (2021) Sequential targeting: a continual learning approach for data imbalance in text classification. *Expert Syst Appl* 179(115067):0957–4174
9. Lee JH, Park KH (2021) GAN-based imbalanced data intrusion detection system. *Pers Ubiquitous Comput* 25(1):121–128
10. Banerjee A, Bhattacharjee M, Ghosh K et al (2020) Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media. *Multimedia Tools Appl* 79:35995–36031
11. Shafqat W, Byun YC (2022) A hybrid GAN-based approach to solve imbalanced data problem in recommendation systems. *IEEE Access* 10:11036–11047
12. Yafooz WMS, Alsaedi A (2021) Sentimental analysis on health-related information with improving model performance using machine learning. *J Comput Sci* 17(2):112–122
13. Suh S, Lukowicz P, Lee YO, Lee H (2021) CEGAN: classification enhancement generative adversarial networks for unraveling data imbalance problems. *Neural Netw* 133:69–86
14. Imran ZKSMDSSAS, Yang R (2022) The impact of synthetic text generation for sentiment analysis using GAN based models. *Egypt Inf J* 23:547–557
15. Mollas I, Chrysopoulou Z, Karlos S, Tsoumacas G (2022) ETHOS: a multi-label hate speech detection dataset. *Complex Intell Syst* 8:4663–4678
16. Devi D, Biswas SK, Purkayastha B (2019) Learning in presence of class imbalance and class overlapping by using one-class SVM and undersampling technique. *Connection Sci* 31(2):105–142

17. Zhang J, Chen L (2019) Clustering-based undersampling with random over sampling examples and support vector machine for imbalanced classification of breast cancer diagnosis. *Comput Assist Surg* 24:62–72
18. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The Weka data mining software: an update. *ACM SIGKDD Explor Newsletter* 11(1):10–18
19. Das AK, Chakrabarty S, Pati SK, Sahaji AH (2012) Applying restrained genetic algorithm for attribute reduction using attribute dependency and discernibility matrix. In: International conference on information processing. Springer, pp 299–308
20. Basu S, Das S, Ghatak S, Das AK (2017) Strength Pareto evolutionary algorithm based gene subset selection. In: 2017 International conference on big data analytics and computational intelligence (ICBDAC). IEEE, pp 79–85
21. Rokach L (2009) A survey of clustering algorithms. In: Data mining and knowledge discovery handbook. Springer, pp 269–298
22. Das P, Das AK (2017) Behavioural analysis of crime against women using a graph based clustering approach. In: 2017 International conference on computer communication and informatics (ICCCI). IEEE, pp 1–6
23. Yasmin G, Chowdhury S, Nayak J, Das P, Das AK (2021) Key moment extraction for designing an agglomerative clustering algorithm-based video summarization framework. In: Neural computing and applications, pp 1–22

# Gender and Hand Identification Based on Dactyloscopy Using Deep Convolutional Neural Network



Diptadip Maiti and Debasish Das

**Abstract** Fingerprint is generally used to accurately identify and authenticate a person due to its distinctive and individual characteristics. Moreover, fingerprint can also describe the person's gender along with the corresponding hand from which the print has been taken. In forensic anthropology, such usage of gender and hand identification from fingerprints is considered as a crucial step. It is taken care to identify a criminal and cut down on the number of suspects being searched for. In this paper, we bring forward a deep ConvNet which is able to categorize fingerprints by individual gender and hand. The deep network achieves a validation accuracy of 99.40% and 99.17% for classification of gender and hand, respectively. The practicality of the proposed network is tested using publicly available SOCOFing data set, which acts as a standard in the result of the categorization technique.

**Keywords** Bio-metric · Fingerprint · Deep ConvNet · Hand identification · Gender recognition

## 1 Introduction

Recognition and validation techniques have become a crucial mechanism in today's world of rising significance for safety in an organization. The need for trustworthy personal identification has led to a rise in interest in biometrics in computerized access control systems [1]. Biometrics are bodily or behavioural characteristics of individuals that can be utilized to precisely point out a human and allow admittance to a system, device, or data. These biometric identifiers comprise fingerprints, facial motif, speech patterns, together with typing cadence [2]. As fingerprints have

---

D. Maiti ( ) · D. Das

Department of Computer Science and Engineering, Techno India University, EM-4, Saltlake, West Bengal 700091, India  
e-mail: [diptadipmaiti@gmail.com](mailto:diptadipmaiti@gmail.com)

D. Das

e-mail: [debasish.d@technoindiaeducation.com](mailto:debasish.d@technoindiaeducation.com)

advantages over other biometric authentication methods, including being practicable, distinct from other fingerprints, permanent, accurate, and widely accepted for use in security and human identification [3]. In legal proceedings around the world, fingerprints are regarded as valid proof of evidence [4]. Dactyloscopy is the inspection of the lines on the fingertips to identify a person. In forensic anthropology, gender and hand classification with fingerprints is a pivotal stride in order to pinpoint a criminal and cut down on the number of suspects that need to be looked into [5]. A growing variety of applications in human–computer interaction (HCI), human–robot interaction, and surveillance technologies can be benefitted from automatic gender and hand identification of people [6]. Moreover, it can help the individual identification process by reducing the possibility of comparisons to one-fourth of the database [7].

In this manuscript, we propose a deep convolution model to identify gender and hand from fingerprint and communicate benchmark result on publicly available SOCOFing data set. The proposed model involves four layers of convolution, accompanied by feature extraction using batch normalization along with maxpooling. Now, the obtained features are fed to the two dense layer artificial neural network classifier which finally identifies the gender and hand of the query fingerprint. The motivation of employing CNN is its robustness property against nonlinear deformation that frequently occurs in the raw fingerprint images during acquisition phase. Moreover, the convolution layer applied different filters in different direction to extract meaningful features from the fingerprint image which results more accuracy in classification and identification phases. The manuscript is assembled in the following manner: Sect. 2 reports the existing literary works on the domain. The proposed methodology and details of data set used for training and testing are described in Sect. 3. Analysis, discussion, and highlights of the pay-off of the designed method are presented in Sect. 4. Conclusion along with future directives is drawn in Sect. 5.

## 2 Related Work

There had been research works carried out on fingerprint identification and classification in recent past. However, very few research has focused at gender and hand recognition and validation through fingerprints. Badawi and Mahfouz [8] developed a fingerprint-based gender classification system, and that is a critical milestone of forensic anthropology. A set of ten fingerprint images from two thousand two hundred persons of diverse ages as well as genders (one thousand one hundred men with one thousand one hundred women) were analysed. They have taken into account the ration of ridge and valley count with their thickness. They also incorporate other features like count of white line, concordance of patterns, and the asymmetric of ridge lines. Data was categorized using fuzzy C-Means (FCM), linear discriminant analysis (LDA) as well as neural networks , each of which produced findings of 80.39%, 85.5%, and 88.5%, respectively. The method uses non-conventional features other than minutiae for detection of gender and study different machine learning methods for achieving better accuracy. Shehu et. al. [9] used deep convolution network for

gender and hand classification with a success rate of 75.2% and 93.5%, respectively. They use transfer learning by taking help of already trained ResNet for gender and hand identification. The data set which contains images of thumbs of right hand and left hand is managed in the searching stage by Mishra and Maheshwary [10], and they were able to cut search time with the use of SVM and Naive Bayes classifiers. The training time and prediction time with SVM and Naive Baye is very fast. Prabha et al. [11] suggested a gender classification approach from fingerprints where the features were drawout utilizing discrete wavelet transform. They used traditional computing methods for gender identification. Gornale et al. [12] established another cutting-edge approach for gender recognition by employing the classifiers KNN and LBP. They obtained the gender detection rate of 95.88% for a set of male and female fingerprint images. With these methods they have achieved better accuracy than their previous method [12]. Falohun et al. [13] developed a system to approximate a individual's age and gender from fingerprint where the training of the machine is done by DWT and PCA features (for age identification). The ridge thickness–valley thickness ratio features were considered to categorize the gender with back propagation neural network. With the help of PCA, they used a reduced feature map for classification of gender. Shinde and Annadate [14] employed SVD along with DWT to determine a human's gender from a fingerprint where the images were investigated up to six levels of DWT along with SVD independently in the study. The comprehensive accuracy percentage for both male with female trial participants was 78.65%. Singular value decomposition is used for dimension reduction of the feature set. It is disclosed by Terhorst and Damer [15] that singular point of fingerprints is greatly connected with the gender and that can be merged with the identification procedure to achieve greater accuracy. To represent every fingerprint in the database, Sarath et al. [16] calculate the ration between ridge and valley thickness with the density of ridge to from a feature vector. They created a reliable classification task for masculine and feminine feature vector motifs by SVD to train on a batch of one hundred fifty male and one hundred twenty five female fingerprints. The feature space had been vectorized, thereby substantially speeds up the execution of the algorithm. Merkel et al. [17] recorded fingerprint pictures using capturing devices (CWL and CLSM) that span ten distinct data sets with a sum of two thousand six hundred eighteen time-series. Machine learning techniques were used to find the age for the whole data set by including coefficients of correlation and age estimation. They tried to estimate the age of a person from the fingerprint image. Shivanand Gornale et al. [18] classified fingerprints into male and female groups using Gabor and DWT-based criteria. For the system's training, fingerprint images of 74 people of various ages and genders were collected. Rekha et al. [19] applied the Gabor filter along with the SVM to distinguish gender from fingerprints. Suman Sahu et al. [20] considered the Adaptive Neuro-Fuzzy Inference System (ANFIS) technique to determine fingerprint gender. For the task of classify fingerprint pictures either female or male, frequency domain examination (frequency-based attributes such as amplitude, perpendicular, parallel, and diagonal) with RVA had been retrieved. They took the advantage of frequency domain analysis and soft computing technique for gender classification.

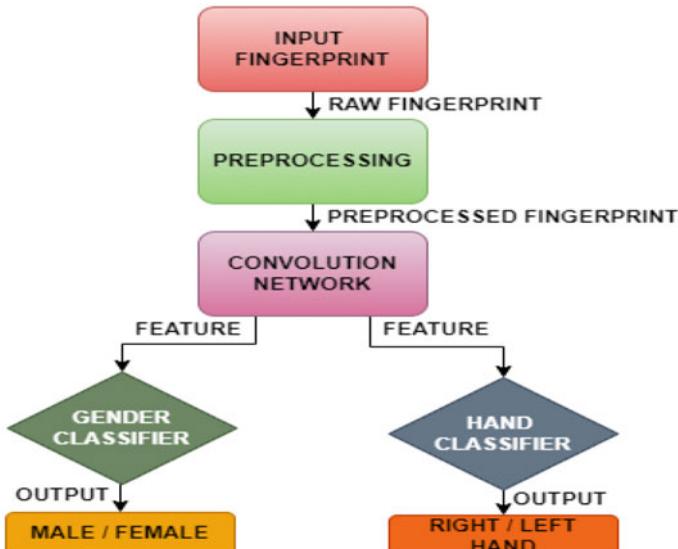
### 3 Proposed Method

We have proposed a CNN architecture to classify human's fingerprint with its corresponding hand and gender. The proposed model has been experimented on a certain raw fingerprint database. The raw data have been pre-processed and fed to a CNN to achieve specific features which have been passed through different two classifiers to obtain the final conclusion. The workflow of the suggested approach is illustrated in Fig. 1.

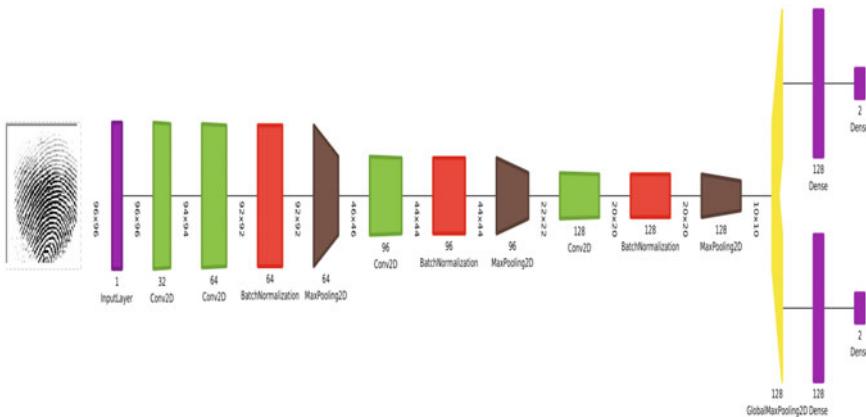
**Network Architecture** To extract features from the input fingerprint, a four convolution layers have been deployed followed by batch normalization and maxpooling. The required filters of the convolution layer and number of nodes in the dense layers have been fixed with the help of rigorous testing with the data set and applying different combination of the hyperparameter of the network to achieve a better accuracy. By following the convolution technique, the final extracted feature is acquired through a global maxpooling operation. The extracted features are fed into two parallel artificial neural networks with 128 hidden nodes each which classify the features as male or female and right or left hand. Figure 2 depicts the proposed model's network architecture.

**Network Hyperparameter** Here, we have enlisted the hyperparameters utilized in the proposed network for gender and hand identification:

- **Activation Function(convolution):** relu
- **Kernel Initializer:** he\_uniform



**Fig. 1** Proposed methods' workflow



**Fig. 2** Proposed network architecture

- **Optimizer:** Adam
- **Activation Function(Dense):** softmax
- **Global Loss Function:** categorical\_crossentropy

Figure 3 represents the details of the proposed model.

## 4 Experimental Results and Performance Analysis

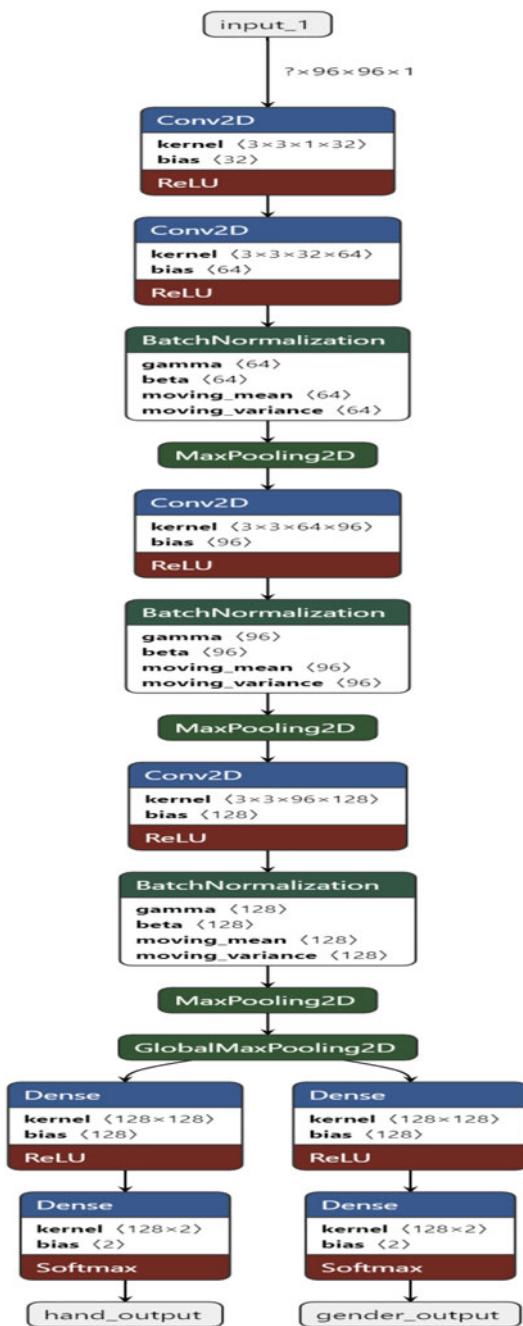
The suggested approach is trained and tested on a publically accessible database called as Sokoto Conentry Fingerprint Database [21]. The data set includes six thousand fingerprint samples of real people that have undergone editing techniques such as obliteration, central rotation, and z-cutting to generate changed easy, altered medium, and altered hard fingerprint images. Figure 4 represents the elements of the database.

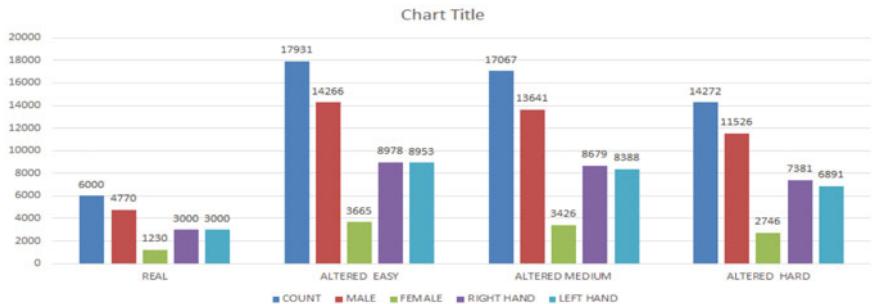
Prior to training and testing, the  $96 \times 103$  resolution raw input fingerprint is scaled to  $96 \times 96$ . The images are then normalized to grey images with minimum value of 0 and maximum value of 1. The complete data set is divided into three sets, with 24,142 images being used for training, 10,347 images being used for validation, and 14,781 images being used for testing.

To achieve the higher accuracy level in gender and hand identification with fingerprint, the network was taught for 100 epochs with a 64-person batch size. The full data set's training component requires the network to train for about 12h. Tables 1 and 2 display the accuracy, loss value, and training with validation results in case of gender and hand classification, respectively.

Tables 3 and 4 exhibit the comparisons between the suggested technique and state of the art relevant works with regard to classification performance for gender and hand identification.

**Fig. 3** Proposed network parameter details



**Fig. 4** Particulars of database SOCOFIG**Table 1** Accuracy with loss for gender categorization during training and testing

Gender recognition	Accuracy (%)	Loss
Training	99.97	0.0012
Validation	99.40	0.0262

**Table 2** Accuracy with loss for hand categorization during training and testing

Hand identification	Accuracy (%)	Loss
Training	99.90	0.0030
Validation	99.17	0.0229

**Table 3** Comparison of suggested procedure with the existing latest practices for gender recognition

S. No.	Title	Technique	Result (%)
1	Method in [8]	Neural network	88.5
2	Method in [11]	Back propagation neural network	96.6
3	Method in [12]	Quantitative descriptive analysis	95.8
4	Method in [13]	Artificial neural network	80
6	Method in [14]	Discrete wavelet transform	78.6
5	Method in [18]	K-nearest neighbours	95.8
7	Suggested method	ConvNet	99.40

**Table 4** Comparison of suggested procedure with the existing latest practices for hand identification

S. No.	Title	Applied method	Applied model	Data set	Accuracy (%)
1	Method in [9]	ConvNet	ResNet	SOCOFing	88.5
2	Suggested method	ConvNet	Custom model	SOCOFing	99.17

**Table 5** Classification report for gender identification

	Precision	Recall	F1-Score	Support
Male	1.00	1.00	1.00	99
Female	1.00	1.00	1.00	29
Accuracy			1.00	128
Macro average	1.00	1.00	1.00	128
Weighted average	1.00	1.00	1.00	128

**Table 6** Classification report for hand identification

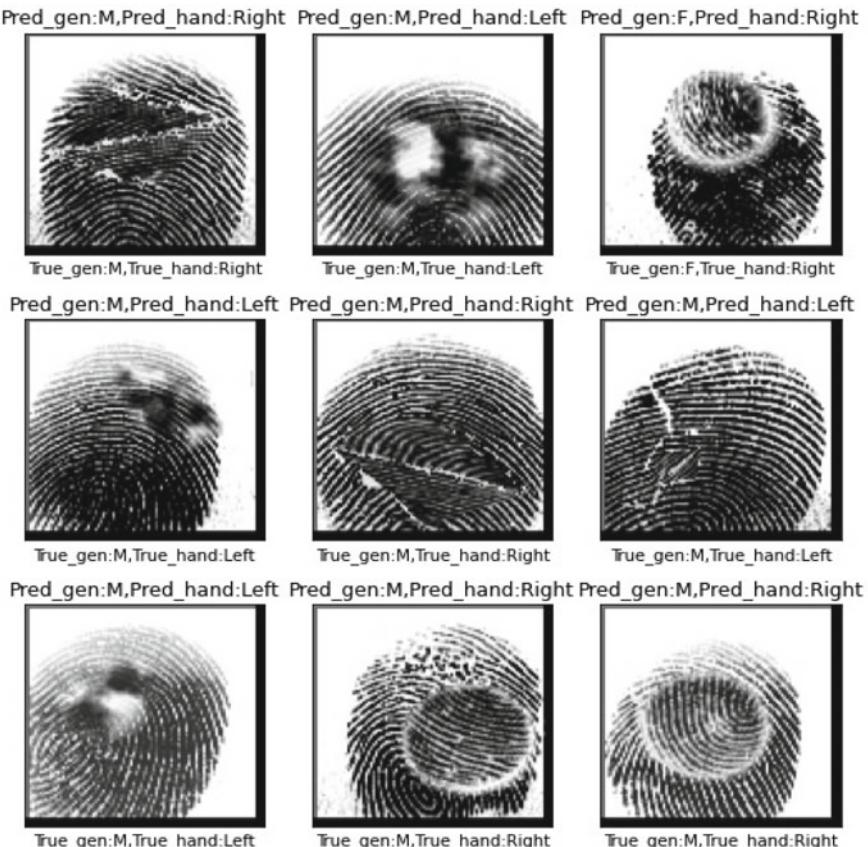
	Precision	Recall	F1-Score	Support
Right hand	1.00	1.00	1.00	58
Left hand	1.00	1.00	1.00	70
Accuracy			1.00	128
Macro average	1.00	1.00	1.00	128
Weighted average	1.00	1.00	1.00	128

The classification report of the proposed network for gender and hand is provided in Tables 5 and 6.

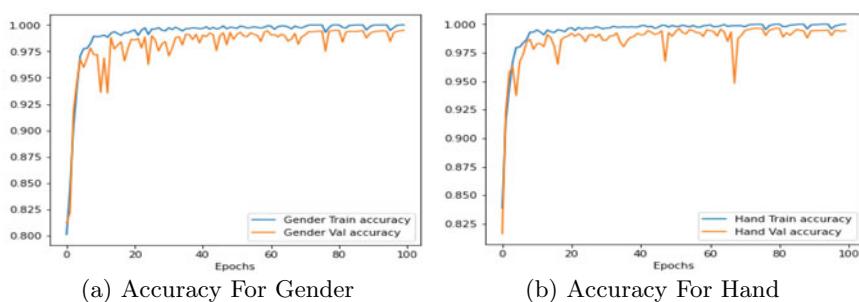
Receiver operating characteristic (RoC) curves for the proposed model for gender and hand recognition are shown in Fig. 7d and f, along with its training and testing accuracy, loss, precession with recall in , Figs. 6a–b, and 7a–d, respectively. A few random sample out-turn of the suggested methodology is exhibited in Fig. 5.

## 5 Conclusion

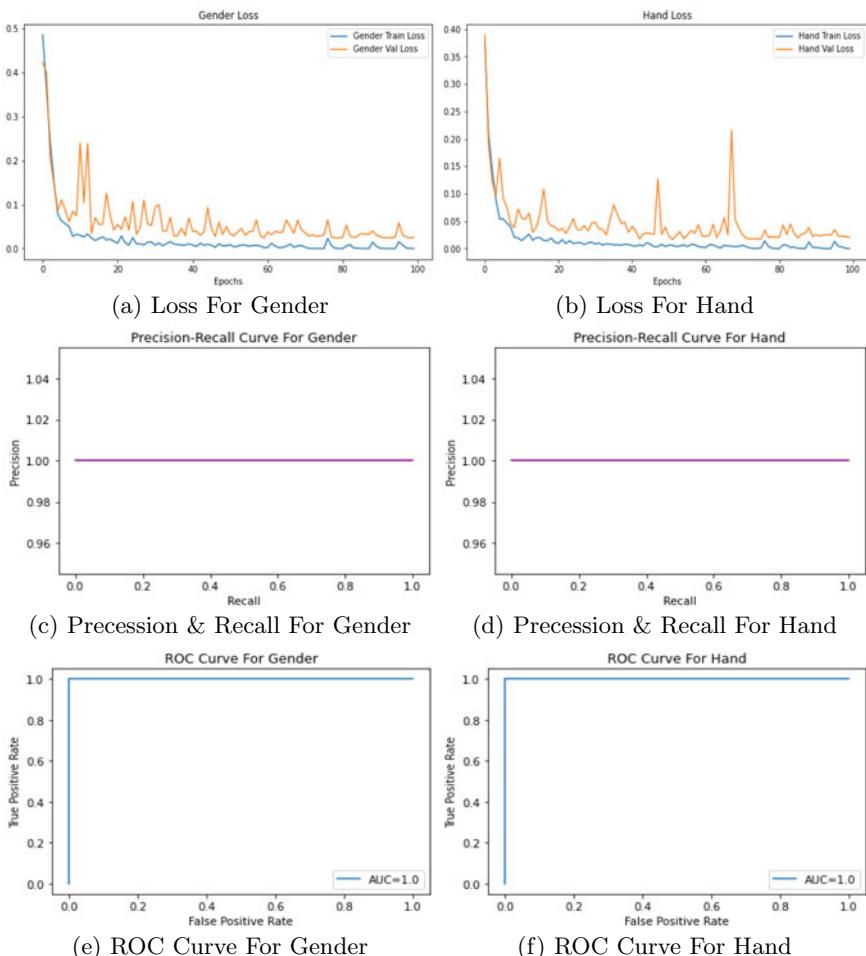
In consequence of the extensive possibilities of fingerprints as a powerful form of recognition, the current study attempted to investigate the relationship between individual's fingerprints and gender or corresponding hand. The findings of gender and hand categorization using these prominent attributes showed that the proposed approach might be useful in the course of forensic anthropology towards cut down the checklist of possible suspects that provide a likelihood value for a suspect's recognition. In future, We'll try to enhance gender and hand categorization by combining fingerprint characteristics from convolution layers with tiny points from the fully connected layer. We shall compare the proposed technique with other classifiers as well. We shall also investigate whether additional parameters, namely fingerprint thickness with valley thickness improves the classification capacity of deep ConvNet.



**Fig. 5** Sample output of the proposed technique



**Fig. 6** Accuracy of gender and hand identification of the proposed model



**Fig. 7** Performance of the proposed network

## References

1. Khan SA, Ahmad M, Nazir M, Riaz N (2013) A comparative analysis of gender classification techniques. *Int J Bio-Sci Bio-Technol* 5(4):223–243
2. Nithin MD, Manjunatha B, Preethi DS, Balaraj BM (2011) Gender differentiation by finger ridge count among south indian population. *J Forensic Legal Med* 18(2):79–81
3. Suwarno S, Santosa PI (2017) A short review of gender classification based on fingerprint using wavelet transform. *Int J Adv Comput Sci Appl* 8(11)
4. Nayak VC, Rastogi P, Kanchan T, Yoganarasimha K, Kumar GP, Menezes RG (2010) Sex differences from fingerprint ridge density in Chinese and Malaysian population. *Forensic Sci Int* 197(1–3):67–69
5. Marasco E, Lugini L, Cukic B (2014) Exploiting quality and texture features to estimate age and gender from fingerprints. In: Biometric and surveillance technology for human and activity

- identification XI, vol 9075, pp 112–121. SPIE
- 6. Gupta S, Rao AP (2014) Fingerprint based gender classification using discrete wavelet transform and artificial neural network. *Int J Comput Sci Mobile Comput* 3(4):1289–1296
  - 7. Divate CP, Ali SZ (2018) Study of different bio-metric based gender classification systems. In: 2018 International conference on inventive research in computing applications (ICIRCA), pp 347–353. IEEE
  - 8. Badawi AM, Mahfouz M, Tadross R, Jantz R (2006) Fingerprint-based gender classification. *IPCV* 6(8):1
  - 9. Shehu YI, Ruiz-Garcia A, Palade V, James A (2018) Detailed identification of fingerprints using convolutional neural networks. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA), pp 1161–1165. IEEE
  - 10. Ashish M, Preeti M (2017) A novel technique for fingerprint classification based on naive bayes classifier and support vector machine. *Int J Comput Appl* 975:8887
  - 11. Jitendra S, Rajmohan P et al (2017) Fingerprint based automatic human gender identification. *Int J Comput Appl* 170(7):1–4
  - 12. Gornale SS, Basavanna M, Kruti R (2017) Fingerprint based gender classification using local binary pattern. *Int J Comput Intell Res ISSN* 0973–1873
  - 13. Falohun AS, Fenwa OD, Ajala FA (2016) A fingerprint-based age and gender detector system using fingerprint pattern analysis. *Int J Comput Appl* 136(4):0975–8887
  - 14. Shinde MK, Annadate SA (2015) Analysis of fingerprint image for gender classification or identification: using wavelet transform and singular value decomposition. In: 2015 International conference on computing communication control and automation, pp 650–654. IEEE
  - 15. Philipp Terhörst, Damer N, Braun A, Kuijper A (2018) Deep and multi-algorithmic gender classification of single fingerprint minutiae. In: 2018 21st International conference on information fusion (FUSION), pp 2113–2120. IEEE
  - 16. Arun KS, Sarath KS (2011) A machine learning approach for fingerprint based gender identification. In: 2011 IEEE recent advances in intelligent computational systems, pp 163–167. IEEE, 2011
  - 17. Ronny M, Jana D, Claus V (2017) A first public research collection of high-resolution latent fingerprint time series for short-and long-term print age estimation. *IEEE Trans Inf Forensics Secur* 12(10):2276–2291
  - 18. Gornale S, Patil A, Veersheety C (2016) Fingerprint based gender identification using discrete wavelet transform and gabor filters. *Int J Comput Appl* 975:8887
  - 19. Rekha V, Gurupriya S, Gayadhri S, Sowmya S (2019) Dactyloscopy based gender classification using machine learning. In: 2019 IEEE international conference on system, computation, automation and networking (ICSCAN), pp 1–5. IEEE
  - 20. Sahu A, Rao AP, Mishra ST. Fingerprints based gender classification using adaptive neuro fuzzy inference system. In: 2015 International conference on communications and signal processing (ICCSP), pp 1218–1222. IEEE
  - 21. Shehu YI, Ruiz-Garcia A, Palade V, James A (2018) Sokoto coventry fingerprint dataset. *arXiv preprint arXiv:1807.10609*

# A CNN-Based Approach for Face Recognition Under Different Orientations



R. Ahila Priyadharshini, S. Hariharan, and R. Jagadeeswara

**Abstract** Face recognition is a difficult task in the realm of computer vision and image analysis. A face recognition model can recognize a face in an image automatically. When the face is orientated in different angles, most of the face recognition systems fail to identify the face, resulting in the degradation of the performance of the facial recognition system. To handle this issue, dataset containing face images should be created in such a manner that it should contain the facial images at varying angles. Face recognition systems based on convolutional neural networks (CNNs) have gained popularity because of its effectiveness in delivering authenticity by automatically identifying faces. The proposed system consists of a robust CNN model for detecting and recognizing the face at varying angles. The proposed CNN model is tested on three datasets, namely ‘MepcoECE’ dataset which was created with 24 subjects in real-time which consists of facial images at varying angles, and two benchmark datasets, namely ORL dataset and GTF dataset, and achieved a superior performance in benchmark datasets.

**Keywords** Face recognition · Data augmentation · Deep learning · CNN

## 1 Introduction

Biometric systems, being a newer technology, may provide a lot of ease by replacing passwords. Facial recognition is a common biometric technique that is frequently utilized in surveillance, security, and criminal identification, among other applications. Face recognition from many angles offers a wide range of applications in our daily lives. Deep learning’s growth has paved the way for standard machine learning algorithms to solve a slew of issues previously thought to be impossible to solve. By generating efficient results, CNNs have changed the way of researchers’ approach

---

R. Ahila Priyadharshini (✉) · S. Hariharan · R. Jagadeeswara

Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi 626005, India  
e-mail: [rahila@mepcoeng.ac.in](mailto:rahila@mepcoeng.ac.in)

to solve the difficulties in computer vision and image processing. CNN-based facial recognition systems have grown in popularity as a result of their ability to give authenticity by automatically identifying faces [1]. The images are seen by CNNs as volumes, or three-dimensional objects with width, height, and depth. Convolutional layers, subsampling layers, and fully connected layers are the basic building blocks of CNNs [2].

The CNN has gained a lot of interest because of its generalizability in solving computer vision issues including continuous emotional dimension prediction [3], plant disease recognition [4], facial micro-expression recognition [5], emotion recognition [6], medicinal plant recognition [7], and so on. CNN designs have developed throughout time to meet difficulties in computer vision. LeNet [8], AlexNet [9], VGG16 [10], and others are examples of popular CNNs.

The tremendous increase in deep learning approaches has overcome the usage of traditional approaches [11–13] for face recognition. The InceptionV2 block and its descendent receptive field block (RFB) are used for increasing the feature map's receptive field by exploiting its numerous branches to address the face recognizing challenge [14]. Yin and Liu [15] presented a multi-task CNN by grouping postures to learn pose-specific identifiers for face recognition. To avoid the time-consuming procedure of explicit feature extraction in classic facial expression recognition, a CNN-based technique based on image edge detection is proposed [16].

In [17], DeepWTPCA-L1, a robust face recognition technique based on CNN-LSTM architecture and WTPCA-L1 features, is proposed. To calculate the characteristics of the proposed method, benchmark databases such as ORL face dataset and GTFD face dataset have been used. With the proposed model, the accuracy of 99.85 and 93.89% is obtained for ORL and GTFD face dataset, respectively. Among the various biometrics, person recognition based on face is very much suitable for real-time applications. Because, lot of features can be extracted from the face, when compared to other biometrics. Challenges that restrict a facial recognition system's potential are illumination, pose variations, low resolution, occlusion, and expression. As a result, many models have trouble identifying faces from different perspectives. In order to handle the pose variation conditions, one possible way is to collect the samples in varying angles of the head position. In this work, we intend to introduce a new face dataset containing faces at different orientations (MepcoECE Dataset) and to recognize the faces at varying angles using a seven layer CNN architecture. The efficiency for the proposed architecture is tested and studied using two benchmark datasets, named ORL face dataset and GTFD face dataset.

## 2 Materials and Method

In this work, we have used two standard datasets and a custom designed dataset for face recognition. The brief details about the dataset and the proposed methodologies are given in this section.



**Fig. 1** Samples in ORL face dataset

## 2.1 *Olivetti Research Laboratory Dataset (ORL Dataset)*

There are ten images for each of the forty unique subjects. For several subjects, images were captured at innumerable times, resulting in varied lighting, face expressions (smiles, frowns, and closed/open eyes), and facial specifics (glasses/no glasses). All the images were captured against a uniformly dark background, with the people facing front [18]. The example images in the ORL database are given in Fig. 1.

## 2.2 *Georgia Tech Face Dataset*

The GTF dataset has face images of 50 persons [19]. All individuals have 15 images, and those images are captured with diverse pose variations, facial expressions, rotations, and numerous lighting conditions. Images are cropped to get the face of the person. So, totally, the database contains 750 images of resolution  $50 \times 50$ . The samples in the GTF dataset are depicted in Fig. 2.



**Fig. 2** Sample images in GTF database



**Fig. 3** Samples of the MepcoECE dataset

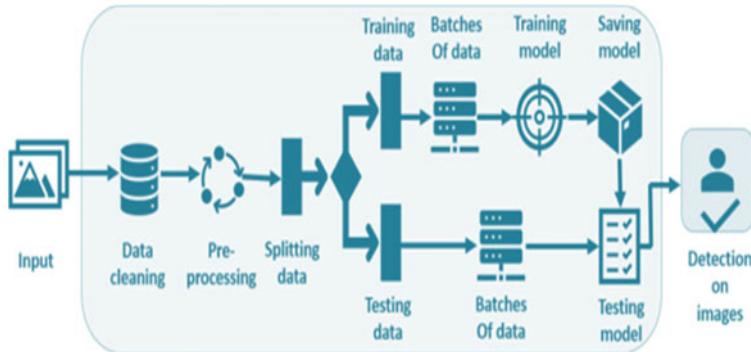
### 2.3 *MepcoECE Dataset*

This dataset is created with 24 persons (subjects). For each person, nearly, 423 to 583 samples were collected. The samples in the dataset are collected in varying angles (180 degree from right to left) of the head position. In order to get only the face part, ‘Haar cascade classifier’ is used [20]. By using this classifier, the face images in the dataset are cropped. The total samples in the dataset are 11,452, and some of the example images of the dataset for the label 20 and label 7 are depicted in Fig. 3.

### 2.4 *Proposed CNN Architecture*

The general flow diagram of face recognition system is presented in Fig. 4. The first step here is to create a dataset. The created dataset contains the facial images with different orientations of different persons. In data cleaning phase, some unwanted samples which is of poor quality will be discarded. In pre-processing stage, data augmentation is done and is for increasing the number of samples. After that the data samples will be splitted into training samples and testing samples. Convolutional neural network model is created to train the samples. In training phase, the features will be extracted from the face image. Once the training process gets completed, the model will be saved. Then in testing phase, the testing samples will be given as input to the model. The model will extract the feature from the test sample and compare it with the features extracted during training phase. The testing sample with features which have maximum matching with the features present in the trained model will be predicted as final output.

The CNN architecture that has been used for the face recognition system is shown in Fig. 5. This architecture consists of seven convolution layers, nine batch normalization (BN) layers, three max pooling layers (MP), and two fully connected layer followed by output layer. The primary convolutional layers extract low-level features such as corners and edges, whereas the final convolutional layer extracts more precise



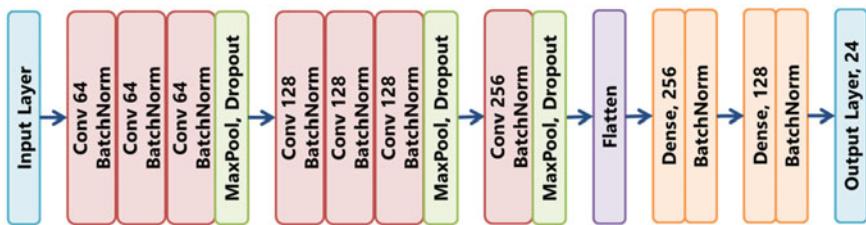
**Fig. 4** General flow diagram of face recognition system

features related to the face images and thereby classify the face images at varying angles.

After each convolutional layer, to enhance the steadiness of the CNN, a batch normalization layer is introduced. The input image is convolved using kernels. Each kernel convolution produces one feature map. The output matrix is also called as feature mapped matrix. All feature maps are stacked to get final output. The output matrix size after convolution operation is given in Eq. 1.

$$O_w = \frac{M - w + 2P_w}{S_w} + 1 \quad O_h = \frac{N - h + 2P_h}{S_h} + 1 \quad (1)$$

where  $O_w \times O_h$  is the size of the output matrix after convolution,  $M \times N$  is the input image size,  $w \times h$  is the kernel size,  $P_w \times P_h$  is width and height of padding,  $S_w \times S_h$  is the stride (number of steps moving across the pixels) applied along width and height during convolution. Every node in the convolution layer has non-linear activation function, named rectified linear unit (ReLU), and is given in Equation 2. This converts non-linearly separable datapoints to linearly separable data points.



**Fig. 5** Proposed CNN architecture

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2)$$

The output of the ReLU is not restricted to a defined range. Normalization is performed shortly before this activation function for preventing unbounded activation from boosting the output layer values. Batch normalization is used in this architecture. It is a trainable layer generally used for addressing the issues of ‘internal covariate shift’. An internal covariate shift happens when the network’s input distribution changes. Hidden layers in the network attempt to learn and adapt to new input distributions as the input distribution changes. The training process is slowed as a result of this. Pooling layer helps to make the output approximately invariant to small translations. Pooling is used to reduce height and width (dimensionality reduction), but the depth remains same. Pooling is done during forward pass, and no updates occur in pooling layer during backpropagation. Adaptive moments (Adam) optimizer is the combination of ‘stochastic gradient descent with momentum optimizer’ and ‘root mean squared propagation’ (RMSPROP). The idea behind the Adam optimizer is to utilize the momentum concept from ‘SGD with momentum’ and adaptive learning rates (LRs) from ‘RMSPROP’. Optimizer is used for minimizing the error function and for maximizing the accuracy. The Adam optimizer has the advantages of being simple to build, computationally efficient, requiring little memory, and being appropriate for problems with extremely noisy/sparse gradients.

A feature vector is created from the image representation after many convolutional layers and down sampling procedures. The fully connected layer receives the feature vector. The final fully connected layer contains same number of neurons as the number of classes to be predicted. The prediction error is computed using a loss function in the output layer. The categorical cross entropy loss function used in this study is depicted in Eq. 3.

$$J(\varphi) = -\frac{1}{N} \sum_{\forall X} \sum_i y_i \log o_i^k + (1 - y_i) \log(1 - o_i^k) \quad (3)$$

where  $N$  represents the number of classes,  $y_i$  and  $o_i$  stand for the actual and predicted labels, respectively, and  $k$  is the layer that corresponds to each class. Following the prediction, the backpropagation learning method modifies the weights and biases for cost and loss reduction.

The training samples are feed forwarded in the CNN model, and the training process takes place. The training also includes some validation data to calculate accuracy of our training. The images that have been splitted for testing phase are then fed into the trained network for prediction. Then, the model will predict the label of the person who is in the test image.

### 3 Experiments and Discussion

The MepcoECE dataset is used to conduct the experiment first. The train test ratio maintained is 70:30. Among 30% of images, 10% are used for validation purpose. After splitting the images present in the dataset into training, validation, and testing set, the proposed CNN model is trained for both augmented and non-augmented samples using ‘Adam optimizer’ with learning rate (LR) as 0.001, loss metric as ‘categorical cross entropy’ and 256 as batch size for different epochs such as 5, 10, and 25 epochs. The training image samples are increased by performing augmentation such as resizing, rotation, and flipping. During training, we infer that the model trained with augmented samples produce good accuracy than the model trained with non-augmented samples. So, the testing is done using the augmented trained model for different epochs, and the corresponding accuracies are mentioned in Table 1. Among those combinations, the good accuracy is obtained for the combination of batch size = 256 and epoch = 25.

#### 3.1 Performance Analysis

Precision is defined as the proportion of accurately predicted positive classes to all anticipated positive data. Recall is defined as the proportion of correctly predicted positive observations to all other positive observations in the actual class. The F1-score is calculated using the precision and recall weighted average. The formulae to find these performance measures are expressed in Eqs. (4–7), respectively. A good model has a high rate of TP and TN while having a low incidence of FP and FN.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

**Table 1** Performance of MepcoECE dataset

Number of epochs	Testing accuracy (%)
5	88
10	97
25	100

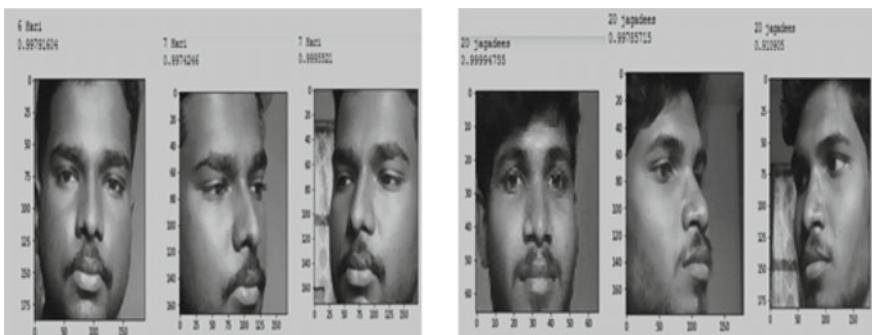


**Fig. 6** Predicted output (OL-original label, PL-predicted label)

The proposed CNN model's predictions using test images produce imperial results. The predicted output of the model is depicted in Fig. 6.

### 3.2 Real-Time Testing

In order to assess the performance of the proposed network, the experiment is further done on real-time images. To prove the robustness of the proposed architecture, the real-time videos of the corresponding subjects are captured and are converted into image frames and classified. The output predicted on real-time samples is shown in Fig. 7, and their performance is shown in Table 2. The confusion matrix for the real-time images is shown in Fig. 8. The overall accuracy is 94.893%.



**Fig. 7** Predicted output for real-time samples

**Table 2** Performance measures for real-time samples

Label	No. of samples	No. of samples predicted correctly	Precision	Recall	F1-score
0	19	14	1	0.74	0.85
1	25	25	1	1	1.00
2	20	20	0.83	1	0.91
3	22	22	0.71	1	0.83
4	19	13	1	0.68	0.81
5	16	16	1	1	1.00
6	20	0	1	1	1.00
7	27	0	0.77	1	0.87
8	17	1	1	0.94	0.97
9	18	18	1	1	1.00
10	21	21	1	1	1.00
11	12	12	1	1	1.00
12	27	26	1	0.97	0.98
13	17	15	1	0.88	0.94
14	29	28	0.93	0.97	0.95
15	22	22	1	1	1.00
16	19	19	1	1	1.00
17	20	19	1	0.95	0.97
18	25	25	1	1	1.00
19	31	31	1	1	1.00
20	24	22	0.96	0.92	0.94
21	19	17	1	0.89	0.94
22	20	19	1	0.95	0.97
23	27	24	1	0.88	0.94

### 3.3 Performance of Proposed Method on Various Bench Mark Databases

To examine the efficiency of the proposed method, the experimentation is repeated using two different benchmark datasets such as ORL dataset and Georgia Tech Face dataset. The samples in the ORL database and GTF database are trained using the proposed model with batch size of 256 and epochs as 25 to get better accuracy. The proposed architecture yields 100% accuracy for both the datasets. The performance comparison of the proposed architecture with existing methods is shown in Table 3, and the performance of our method is comparable the existing approaches.

Actual label	Predicted label																										
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23			
labels	0	14	0	1	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
1	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	0	0	2	0	13	0	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0		
5	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
6	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
7	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
8	0	0	0	1	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
9	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
10	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
11	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0		
12	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	1	0	0	0	0		
13	0	0	0	0	0	0	0	2	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	
17	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	22	0	0	0	0	0	0
21	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	
22	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	
23	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	

**Fig. 8** Confusion matrix for real-time images**Table 3** Performance comparison with existing methods

Methods	Accuracy (%)	
	ORL dataset	GTF dataset
CNN-LSTM [17]	99.85	93.89
Xception + AC [21]	87.5	—
GTF[22]	—	87.71
Proposed method	100	100

## 4 Conclusion

The suggested face recognition model gives very good accuracy in identifying the faces at different angles, according to the experimental study. The paper also provides a comprehensive breakdown of the key components needed to build a solid deep learning model. The effectiveness of the created deep network is investigated by altering the epochs and carrying out augmentation. Overall, the model also does well

with real-time samples. By integrating this application into any mobile or CCTV camera, this work's reach can be increased to benefit real-time security systems.

## References

1. Batbaatar E, Li M, Ryu KH (2019) Semantic-emotion neural network for emotion recognition from text. *IEEE Access* 7:111866–111878
2. Ahila Priyadarshini R, Arivazhagan S, Arun M (2021) A deep learning approach for person identification using ear biometrics. *Appl Intell* 51(4):2161–2172
3. Meng H, Bianchi-Berthouze N, Deng Y, Cheng J, Cosmas JP (2016) Time-delay neural network for continuous emotional dimension prediction from facial expression sequences. *IEEE Trans Cybern* 46(4):916–929
4. Ahila Priyadarshini R, Arivazhagan S, Arun M et al (2019) Maize leaf disease classification using deep convolutional neural networks. *Neural Comput Appl* 31:8887–8895. <https://doi.org/10.1007/s00521-019-04228-3>
5. Feng XU, Zhang J-P (2017) Facial microexpression recognition: a survey. *Acta Automatica Sinica* 43(3):333–348
6. Özerdem MS, Polat H (2017) Emotion recognition based on EEG features in movie clips with channel selection. *Brain Inf* 4(4):241–252
7. Ahila Priyadarshini R, Arivazhagan S, Arun M (2021) Ayurvedic medicinal plants identification: a comparative study on feature extraction methods. *Commun Comput Inf Sci* 1377:268–280
8. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324. <https://doi.org/10.1109/5.726791>
9. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90. <https://doi.org/10.1145/3065386>
10. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*
11. Ghimire D, Jeong S, Lee J, Park SH (2017) Facial expression recognition based on local region specific features and support vector machines. *Multimed Tools Appl* 76(6):7803–7821
12. Arivazhagan S, Priyadarshini RA, Sowmiya S (2014) Face recognition based on local directional number pattern and ANFIS classifier. In: 2014 IEEE International conference on advanced communications, control and computing technologies, pp 1627–1631. <https://doi.org/10.1109/ICACCCT.2014.7019384>
13. Arivazhagan S, Ahila Priyadarshini R, Sowmiya S (2014) Facial expression recognition based on local directional number pattern and ANFIS classifier. In: 2014 International conference on communication and network technologies, pp 62–67. <https://doi.org/10.1109/CNT.2014.7062726>
14. Li X, Yang Z, Wu H (2020) Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks. *IEEE Access* 8:174922–174930. <https://doi.org/10.1109/ACCESS.2020.3023782>
15. Yin X, Liu X (2018) Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Trans Image Process* 27(2):964–975. <https://doi.org/10.1109/TIP.2017.2765830>
16. Zhang H, Jolfaei A, Alazab M (2019) A face emotion recognition method using convolutional neural network and image edge computing. *IEEE Access* 7:159081–159089. <https://doi.org/10.1109/ACCESS.2019.2949741>
17. Maafiri A, Elharrouss O, Rfifi S, Al-Maadeed S, Chougdali K (2021) DeepWTPCA-L1: a new deep face recognition model based on WTPCA-L1 norm features. *IEEE Access* 9:65091–65100. <https://doi.org/10.1109/ACCESS.2021.3076359>

18. Tavares M (2020) The ORL database for training and testing. Kaggle. Retrieved 19 Apr 2023 from <https://www.kaggle.com/datasets/tavarez/the-orl-database-for-training-and-testing>
19. <https://computervisiononline.com/dataset/1105138700>. Retrieved 19 Apr 2023
20. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, Kauai, HI, USA, pp. I-I. <https://doi.org/10.1109/CVPR.2001.990517>
21. Guérin J, Gibaru O, Thiery S, Nyiri E. CNN features are also great at unsupervised classification. <https://doi.org/10.48550/arXiv.1707.01700>
22. Li R, Zhou Z, Liu X, Li D, Yang W, Li S, Liu Q (2021) GTF: An adaptive network anomaly detection method at the network edge. *Secur Commun Netw* 2021: 4510–4520. Article ID 3017797. <https://doi.org/10.1155/2021/3017797.VPR>

# A Deep Learning Approach for Detection of Disease in Plant Leaves



S. Khatri, B. K. Tripathy , K. V. S. S. Kumar, V. Kumar, S. Bharti, A. Gupta, and Y. Khandelwal

**Abstract** India stands in second in the agriculture sector worldwide. The Indian food industry has already expanded its commitment to the world food exchange because of its huge potential. In India, approximately 15–25% of potential yield is lost due to various diseases in crops. A lot of research has been done in this field to tackle the problem, most of them using deep learning (artificial neural networks, support vector machines, etc.). In this paper, after making an extensive literature survey, it was observed that the existing models have some drawbacks. One of these is being that most of the models directly feed the RGB image into the model as input. Therefore, there is no separation of the lightness and the color components of the image. Hence, it was planned to overcome these drawbacks by building a model that separates the lightness and the color components of the image, operating on both of them individually, and is also computationally more effective. First, transfer learning using ResNet, AlexNet, and VGG models is explored, where AlexNet performed the best with 92.02% accuracy. Later, the implementation of an ensemble gave better performance with an accuracy of 94.52%. Again, custom EfficientB0 architecture performed the

---

S. Khatri · B. K. Tripathy · K. V. S. S. Kumar · V. Kumar · S. Bharti · A. Gupta ·

Y. Khandelwal

School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu 632014, India

e-mail: [tripathybk@vit.ac.in](mailto:tripathybk@vit.ac.in)

S. Khatri

e-mail: [Somya.khatri2019@vitstudent.ac.in](mailto:Somya.khatri2019@vitstudent.ac.in)

K. V. S. S. Kumar

e-mail: [Venkata.saisankar2019@vitstudent.ac.in](mailto:Venkata.saisankar2019@vitstudent.ac.in)

V. Kumar

e-mail: [varun.kumar2019a@vitstudent.ac.in](mailto:varun.kumar2019a@vitstudent.ac.in)

S. Bharti

e-mail: [sejal.bharti2019@vitstudent.ac.in](mailto:sejal.bharti2019@vitstudent.ac.in)

A. Gupta

e-mail: [akash.gupta2019@vitstudent.ac.in](mailto:akash.gupta2019@vitstudent.ac.in)

Y. Khandelwal

e-mail: [yash.khandelwal2019@vitstudent.ac.in](mailto:yash.khandelwal2019@vitstudent.ac.in)

better with an accuracy of 98.03%. Finally, multipath CNN InceptionV3 variation gave the highest accuracy of 99.11%.

**Keywords** Deep learning · Transfer learning · Ensemble · Custom EfficientB0 model · Multipath CNN with 2 branches · InceptionV3 ex terms

## 1 Introduction

Artificial intelligence, remote sensing, and blockchain technology are some of the important components of new technologies proposed by government of India under its mission in connection with digital agriculture for the period 2021–2025. The Agriculture Ministry is hopeful that the process of modernization of the agriculture industry will lead to generation of higher revenue by the farmers continually.

Due to India's tremendous agricultural potential, the food business has already increased its commitment to the global food exchange. But since most of the farmers are semi-literate, it is very difficult for these farmers to detect and identify diseases at an early stage and is only identifiable when the crop is already ruined. It is also very difficult to know proper medication that is to be recommended. As a result, approximately 15–25% of potential yield is lost due to various diseases in crops. Such huge scope crop-misfortune is adversely affecting the farming biosafety which is the principle to food security. So, to tackle all these problems by the means of this paper, we will ensure early detection of diseases in a particular crop so that we can provide the possible remedies to aid the farmers.

## 2 Background

A study carried out by UN has made the prediction that although we are self-sufficient in food requirements in 2021 due to the explosion of population, there is every chance of food shortage by the year 2050. So, there is an urgent requirement to find solutions to this anticipated problem. One of the solutions can be to increase the production by protecting the crop from attack of insects and diseases. The work in this paper is a step in this direction. Farmers can get more yields from their fields by using this technique as it deals with prediction of crop diseases [1]. The technique to be followed comes under classification of data and is broadly categorized under machine learning.

Agriculture being a domain having average risk requires that the prediction of yielding of crops should be done accurately in a country like India. Careful and accurate prediction through proper decision-making helps in the management of risk in agriculture. There is a significant difference in the increase of population and increase in the production of food. So, the demand for food production is of utmost importance to meet the food requirements of the exploding population of the world.

In this paper, it is aimed at predicting the diseases occurring in plant leaves at an early stage, so that appropriate action can be taken early, which will save the plants from having severe damage. This will lead to higher production of crops and minimize the loss of farmers. Even under unfavorable conditions, the loss in yield will be minimum. Also, the prediction of crop yield can be used by the managers. The early detection of diseases in the leaves will require much less pesticides or even no pesticide. So, this proposed model will be helpful for farmers in having very less loss in crop yield due to detection and prevention of leaf diseases.

Flask server is used in the proposed model. The Web interface is developed using HTML and CSS, which takes care of prediction. Python language is used for the development of the E-commerce system.

### 3 Objectives

Early identification of crop diseases and effective treatment can save acres of farms. But even an expert may not be confident of detecting this. As a result, it is necessary to develop an autonomous system for detection that too at an early stage.

- This scenario affects the economy of a country besides the decrement in the health status of the plants as there will be lower production. So, this system will help in increasing the overall yield by eliminating crop destruction risks.
- Appropriate solutions will be proposed to the farmer based on the type of disease found using the system.
- Thus, improving the economic condition of farmers and the agricultural sector.

### 4 Literature Review

Flood filling algorithm and an un-identical color space system were developed for leaf region segmentation in [2]. For the identification of the destination color, initial nodes, and a substitute color, a variant of the flood filling algorithm is used. Taking 5 different varieties of crops from Bangladesh, a technique based on computer vision was developed to detect the presence of plant disease in [3]. MobilenetV2 is used here, and data is uploaded into MobileNet feature extraction layers containing convolutional and max pooling layers.

These days, in order to find out the features of leaves and their extent of infection, the artificial neural networks (ANNs) and support vector machine (SVM) models are used extensively [4]. Deep learning (DL) models are being used in many real-life applications nowadays [5–8]. Convolutional neural networks (CNNs) [9] have proved itself to be an efficient model for image classifications. In [10], taking a combination of the 4 models of CNN in the form of ResNet, InceptionNet, DenseNet, and Inception-ResNet, an algorithm for detection of plant disease was proposed. A stacking method is used, and a voting mechanism is added to detect 61 classes

of healthy and disease plant leaves. Selection of proper training settings is very important as DL models are prone to overfitting. For training, a pre-trained DenseNet-121 model partially to handle overfitting, which may occur as an outcome of data augmentation and optimization techniques has been explored in [11]. Here, to train the network for epochs of 20, 30, 40, and 50 size, seven optimizers were used. The continual decay of learning rates is used throughout the training process.

In [12], an effective method is proposed to distinguish healthy plant leaves from those which are unhealthy and find the diseases with levels of severity using SVM and grid search method for optimization. In [13], a DL-based mathematical model for detecting and recognizing plant diseases to improve training efficiency, generality, and accuracy is proposed. Region proposal network (RPN) is used in the beginning for leave identification and finds their location. Using Chan–Vese (CV) method, the output images are segmented, and then, these are given as input to the transfer learning model. This model is already trained in a simple environment through a dataset of unhealthy leaves. This process takes a lot of time in iteration. In [14], information on current potential plant-pathogen diagnostic approaches is consolidated. It also concentrates on the most up-to-date in-field pathogen diagnosis techniques. Nucleic acid amplification test (NAAT) techniques for in-field diagnosis have been reviewed here along with various isothermal amplification techniques like recombinase polymerase amplification (RPA), loop-mediated isothermal amplification (LAMP), and other diagnostic methods, which are useful for in the study of plant biosecurity. In [15], a multi-class classification problem is handled by taking the 18 disease class-based plant village dataset, which is freely accessible. The efficiency of five models; VGG16, ResNet50, InceptionV3, Inception-ResNet, and DenseNet169 are compared, and ResNet50 was found as the best using standard metrics. The images of tomato plants were taken from the Village dataset for classification using pre-trained models like VGG16 and DenseNet and transfer learning. The study in [16] which uses CNN and AI for paddy crops is carried out. However, the diseases cannot be detected in the early stage. In [17], existing classification algorithms are taken for classifying and predicting diseases using soybean and mushroom datasets. The poisonous varieties are considered as diseased, and edible ones are considered as free from disease. The accuracy of all the algorithms except the Naïve Bayes is 100%. For the soybean dataset, it has shown that ANN and KNN are the best classifiers in terms of accuracy. A new era in the study of plant health was started when image processing and ecological sensing were blended with IoT in [18].

Techniques similar to that of [17] are used in [19] for disease detection and classification. It first detects and records the contaminated area and then performs image pre-processing. It then collects the fragments, identifies the infected area, and performs a feature extraction there. A need to develop fast, inexpensive, and reliable disease detection technologies for health monitoring in order to advance agriculture was recognized in [20].

In order to increase products in the field of agriculture a CNN model was introduced in [21] through identification of diseases in plants. As stated earlier, detection of diseases early is instrumental in increasing the production of agricultural products. In [22], a study was carried out to differentiate various fungal and bacterial

diseases in capsicum plants. In order to locate the affected regions in capsicums, clustering methods were used in the form of k-means. After this, the texture in the form of GLCM is eliminated. These characteristics are useful in the determination of fungal and bacterial diseases in capsicum. But, the model developed is useful for only capsicum plants. A robot was proposed for the detection of diseases in plants through processing of images in [23]. Sickness in plants is identified from the images. The steps image division, image generation, grouping, and extraction of brightness are required for the detection of the affected area. After extraction of these features, SMS is sent to the concerned persons, who may be farmers or other cultivators.

#### **4.1 *Gaps in the Literature***

After the extensive literature survey, it was observed that there are many problems and limitations in the existing algorithms. The major problem observed was that in most of the models RGB images are directly fed into the model as input. Therefore, there is no separation of the brightness and the color components of the image. Most of the disease prediction in plant leaves is based on the change of color in different parts of the leaf. Hence, we are not concerned about how dark or how light the color of the leaf is. Hence, we can separate out the brightness component of the image from the leaf.

Another drawback is poor performance of the existing systems when the leaf is physically damaged which is bound to occur. Also, the existing models cannot handle images containing blur, motion blur, occlusion, and other defects which can occur due to defects in the camera.

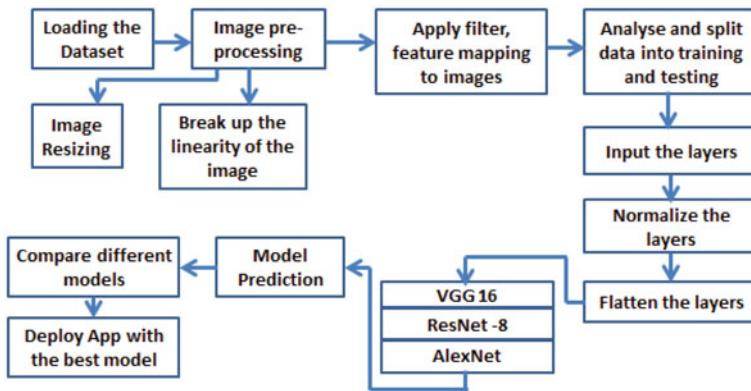
Hence, it is aimed in this work to overcome these drawbacks by building a model that separates the brightness and the color components of the image and operating on both of them individually. Also, it is planned to build a model that can handle noisy data without any major decrease in the model performance and does not show any decrease in accuracy when the leaf is physically damaged.

### **5 System Architecture**

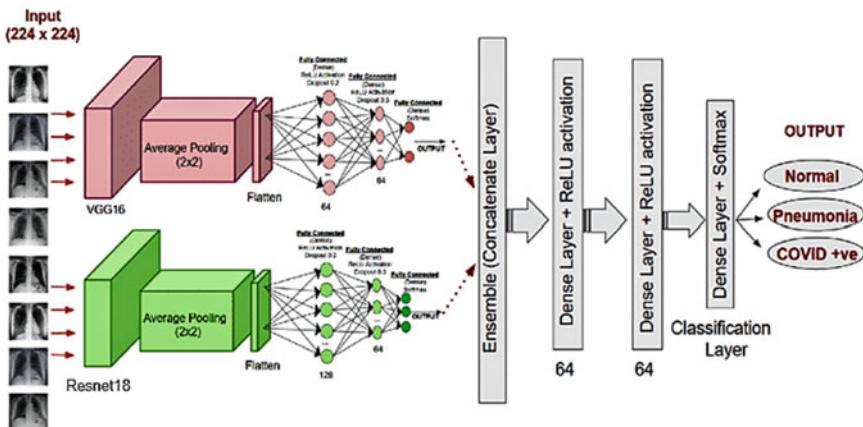
In this section, we provide architectures of different components of the system.

#### **5.1 *Transfer Learning***

Figure 1 shows the architecture of the working process of transfer learning.



**Fig. 1** Transfer learning architecture



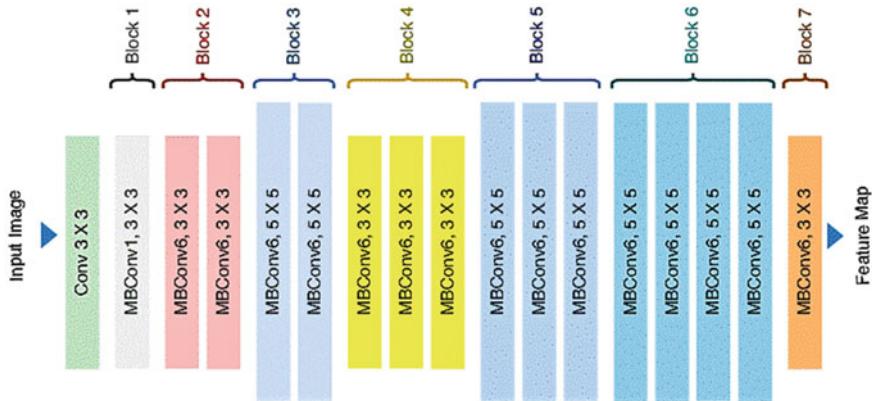
**Fig. 2** Ensemble voting classifier

## 5.2 Ensemble Voting Classifier

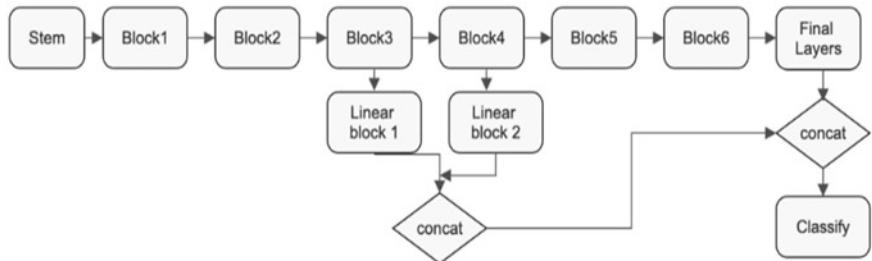
Figure 2 shows the working of the ensemble voting classifier. Figure 3 shows the architecture of the custom EfficientB0 model, and Fig. 4 shows its workflow diagram.

## 5.3 Multipath CNN InceptionV3

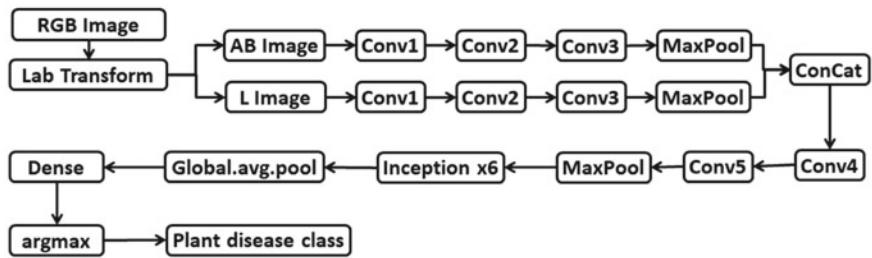
Figure 5 is a flowchart on the working of the multipath CNN InceptionV3 model.



**Fig. 3** Architecture of custom EfficientB0 model



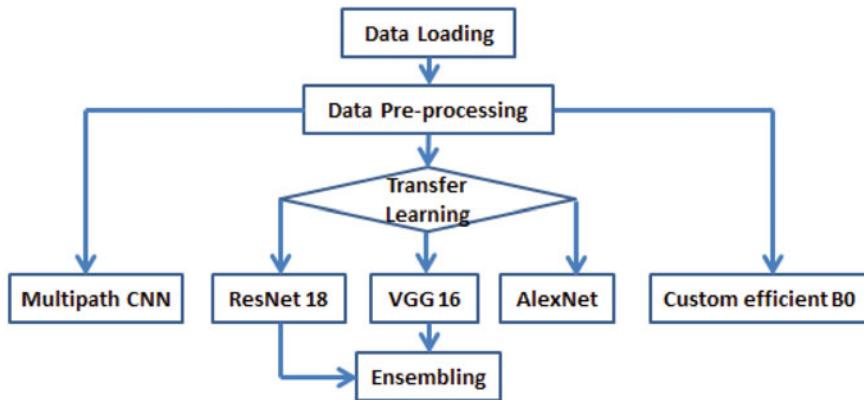
**Fig. 4** Workflow diagram of custom EfficientB0 model



**Fig. 5** Multipath CNN InceptionV3 model

#### 5.4 Flowchart of the Proposed Model

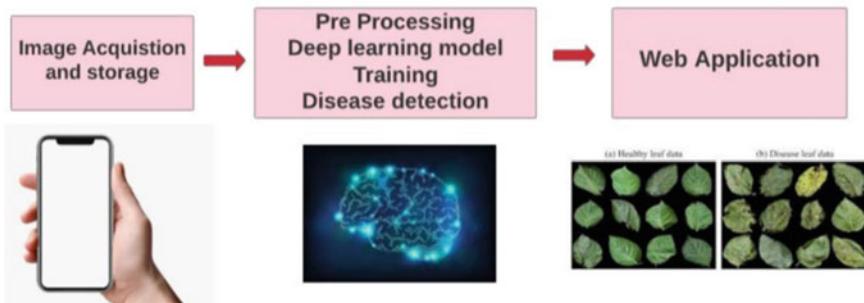
Figure 6 is the flowchart describing the workflow of our proposed model.



**Fig. 6** Flow diagram of the proposed model

### 5.5 Proposed Multipath CNN InceptionV3

This proposed method is a variation of the InceptionV3 architecture of convolutional neural networks. It is a multipath CNN consisting of 2 branches. The input RGB image converted to L\* and AB\* components. The first branch is fed with achromatic L-channel, and the other branch is fed with AB-channel. Both the branches consist of 3 convolutional layers followed by a Max Pooling layer. Then, both the branches are merged together and provided to 2 more convolutional layers and another set of Max Pooling and global average pooling layer, finally giving the disease class as the output class label. This proposed method provides resistance to adverse noise effects on the data. This method provides effective isolation of lightness and color components leading to improved accuracy. This method also provides resistance to noisy data and resistance to physically damaged leaves. The working steps of the proposed model are shown in Fig. 7.



**Fig. 7** Working of proposed model

**Table 1** Various image transformation functions used for image pre-processing

Image transformation function	The role of the functions
transforms.compose	It combines all the transforms provided to it
transforms.RandomResizedCrop	Random part of the image is cropped and resized
transforms.Normalize	Tensor image is normalized
transforms.ToTensor()	Converts a PIL Image to tensor
transforms.CenterCrop	Central region of the image is cropped and resized
transforms.Resize	The input image resized to the given size

## 6 Descriptions of Components and Concepts

In this section, the descriptions of the components and concepts of the proposed model are discussed in detail.

### 6.1 Data Description

Images were used from the plant village dataset from Kaggle dataset link. It contains 38 labels, 54,306 images, 26 diseases, and 14 crop species. We have chosen 5 classes (multiple diseases, healthy, rust, scab, black rot) of leaves containing 15,110 images.

### 6.2 Data Prepossessing

Data size plays a major role in training it, and in order to increase data size, augmentation techniques are used. In this paper, several data augmentation techniques like rescale, fill mode, rotation range, width shift range, zoom image, height shift range, and shear change are used. In Table 1, we describe functionalities of the image pre-processing functions used in this work.

Some of the important concepts used in this work have been described in Table 2.

### 6.3 Custom EfficientB0

It is a computationally efficient model with an accuracy of 84.4% on the ImageNet dataset. Optimization of floating-point operation and accuracy is achieved through an architecture with multiple objectives considered simultaneously. Using B0 as a baseline model, a full family of EfficientNets from B1 to B7 are developed leading to a high accuracy on ImageNet. Our proposed custom model follows a similar process

**Table 2** Model/concepts description

Model/Concept	Description
Data splitting	For transfer learning, ensemble, custom EfficientB0 and multipath CNN, the dataset is divided into 3 different groups; train, validation, and test. Each group is divided into, 80% for training, and 20% for testing
Transfer learning	We use pre-trained weights from the VGG-16, AlexNet, and ResNet18 models as a part of transfer learning which is a well-known deep learning technique
AlexNet	The primary concept used in AlexNet is that for high performance, the depth of the model is essential. Use of GPU made this high complexity model feasible
Visual Geometric Group 16 (VGG 16)	Replacement of kernels with large size filters is replaced with several kernel sized filters of size $3 \times 3$ which is made leading to 92.7% accuracy in this CNN model. In it, 3 fully connected layers follow 13 convolutional layers. It is suitable for smaller network architectures such as SqueezeNet and GoogleNet
Residual Network (ResNet)	Several blocks of CNN are used. It leads to a sequence of CNN blocks, which takes input channels and output channels. Each convolution layer is followed by a batch norm 2D. Using image channels, number of blocks, number of classes, and number of layers, a reset class is created
Ensemble	Ensembling of VGG16 and ResNet18 has been done to make and ensemble vote classifier, and both hard and soft voting has been performed, which showed us better results than individual ones

by taking the EfficientB0 model. In the process, large size information is taken and transformed into minute information to add context.

## 7 Results

In this section, we provide the comparison of the efficiency of the different models, which shows the efficiency of the proposed model (Table 3).

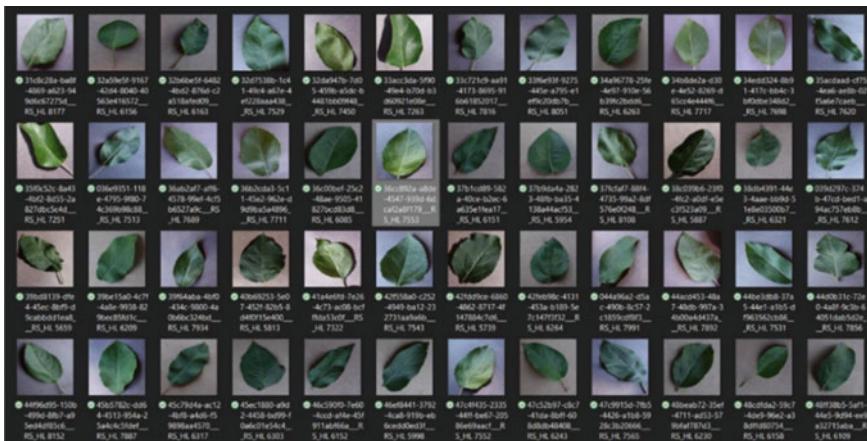
The training dataset is displayed in Fig. 8.

### 7.1 Comparison of Accuracy

The bar graph in Fig. 9 represents the accuracy comparison of different models used in the detection of disease in plants leaves. Multipath CNN InceptionV3 variation model with 20% L and 80% AB has the highest accuracy of 99.16%.

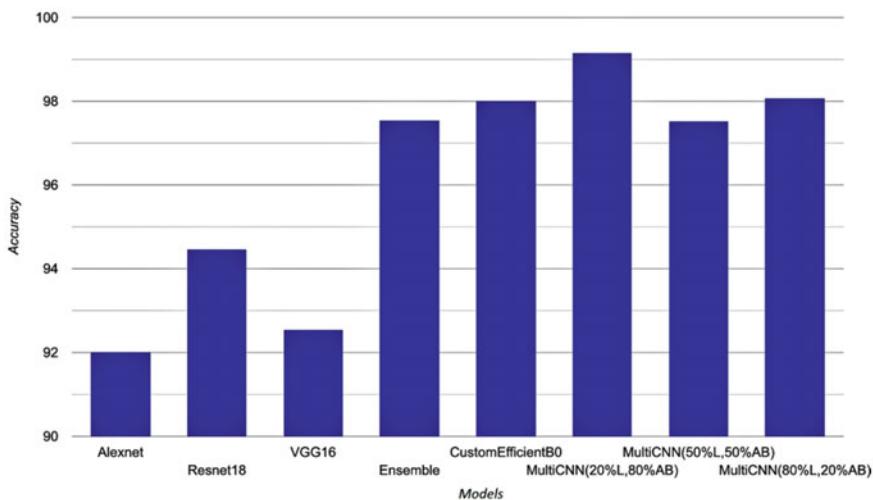
**Table 3** Accuracy of the different models under study

Model	Accuracy
AlexNet	92.02
ResNet18	94.47
VGG16	92.55
ENSEMBLE	97.55
Custom EfficientB0	98.03
Multipath CNN InceptionV3 variation 20% L 80% AB	99.16
Multipath CNN InceptionV3 variation 50% L 50% AB	97.53
Multipath CNN InceptionV3 variation 80% L 20% AB	98.08

**Fig. 8** Training dataset

## 8 Conclusions

Several measures are taken in this work which has provided an improved accuracy than all the competitive models. As the brightness and the color components of the image are separated and operated individually. The performance of the model also improved for the leaves which are physically damaged too. Use of pre-trained models such as VGG16, AlexNet, and ResNet18 improved the efficiency further. For example, the pre-trained weights of the ImageNet dataset assisted in adjusting the restricted number of images per class as well as the computing requirements (transfer learning). Using ensemble of ResNet and VGG16 increased the accuracy further. The system could be protected against considerably high variance or excessive bias due to the use of custom EfficientB0 model. Finally, a variation of the InceptionV3 architecture of CNN is used and the input RGB image converted to L\* and AB\* components, so that the final accuracy achieved came up to as high as 99.11%.



**Fig. 9** Comparison of accuracy of different models

## References

1. Rubini P, Kavitha P (2021) Deep learning model for early prediction of plant disease. In: 2021 Third International conference on intelligent communication technologies and virtual mobile networks (ICICV), pp 1104–1107. <https://doi.org/10.1109/ICICV50876.2021.9388538>
2. Nagamani HS, Devi HS (2021) Leaf region segmentation for plant leaf disease detection using color conversion and flood filling. In: 2021 2nd Global conference for advancement in technology (GCAT). IEEE, pp 1–7
3. Moyazzoma R, Hossain MAA, Anuz MH, Sattar A (2021) Transfer learning approach for plant leaf disease detection using CNN with pre-trained feature extraction method Mobilnetv2. In: 2021 2nd International conference on robotics, electrical and signal processing techniques (ICREST). IEEE, pp 526–529
4. Tripathy BK, Anuradha J (2015) Soft computing—advances and applications. Cengage Learning Publishers, New Delhi. ASIN 8131526194, ISBN: 10-9788131526194
5. Bhattacharyya S, Snasel V, Hassanian AE, Saha S, Tripathy BK (2020) Deep learning research with engineering applications. De Gruyter Publications. ISBN: 3110670909, 9783110670905. <https://doi.org/10.1515/9783110670905>
6. Bose A, Tripathy BK (2020) Deep learning for audio signal classification. In: Bhattacharyya S, Hassanian AE, Saha S, Tripathy BK (eds) Deep learning research and applications. De Gruyter Publications, pp 105–136
7. Kaul D, Raju H, Tripathy BK (2022) Deep learning in healthcare. In: Acharjya DP, Mitra A, Zaman N (eds) Deep learning in data analytics- Recent techniques, practices and applications. Studies in big data, vol 91. Springer, Cham, pp 97–115. [https://doi.org/10.1007/978-3-030-75855-4\\_6](https://doi.org/10.1007/978-3-030-75855-4_6)
8. Adate A, Tripathy BK (2022) A survey on deep learning methodologies of recent applications. In: Acharjya DP, Mitra A, Zaman N (eds) Deep learning in data analytics- recent techniques, practices and applications. Studies in big data, vol 91. Springer, Cham, pp. 145–170. [https://doi.org/10.1007/978-3-030-75855-4\\_9](https://doi.org/10.1007/978-3-030-75855-4_9)
9. Maheswari K, Shaha A, Arya D, Tripathy BK, Rajkumar R (2020) Convolutional neural networks: a bottom-up approach. In: Bhattacharyya S, Hassanian AE, Saha S, Tripathy BK

- (eds) Deep learning research with engineering applications. De Gruyter Publications, pp. 21–50. <https://doi.org/10.1515/9783110670905-002>
- 10. Guan X (2021) A novel method of plant leaf disease detection based on deep learning and convolutional neural network. In: 2021 6th International conference on intelligent computing and signal processing (ICSP). IEEE, pp. 816–819
  - 11. Noon SK, Amjad M, Qureshi MA, Mannan A (2020) Overfitting mitigation analysis in deep learning models for plant leaf disease recognition. In: 2020 IEEE 23rd International multtopic conference (INMIC). IEEE, pp. 1–5
  - 12. Bhagat M, Kumar D, Haque I, Munda HS, Bhagat R (2020) Plant leaf disease classification using grid search based SVM. In: 2nd International conference on data, engineering and applications (IDEA). IEEE, pp 1–6
  - 13. Guo Y, Zhang J, Yin C, Hu X, Zou Y, Xue Z, Wang W (2020) Plant disease identification based on deep learning algorithm in smart farming. Discr Dyn Nat Soc 1–11. <https://doi.org/10.1155/2020/2479172>
  - 14. Mansotra R, Vakhlu J (2021) Comprehensive account of present techniques for in-field plant disease diagnosis. Arch Microbiol 203(9):5309–5320
  - 15. Sagar A, Jacob D (2020) On using transfer learning for plant disease detection. Discr Dyn Nat Soc 2020:1–11
  - 16. Vardhini PAH, Asritha S, Devi YS (2020) Efficient disease detection of paddy crop using CNN. In: 2020 International conference on smart technologies in computing, electrical and electronics (ICSTCEE), pp 116–119. <https://doi.org/10.1109/ICSTCEE49637.2020.9276775>
  - 17. Maria M, Carla B, Raed S (2021) Plant disease prediction using classification algorithms. IAES Int J Artif Intell (IJ-AI) 257–264
  - 18. Yashodha G, Shalini D (2021) An integrated approach for predicting and broadcasting tea leaf disease at early stage using IoT with machine learning—a review. Mater Today Proc 37(Part 2):484–488. <https://doi.org/10.1016/j.matpr.2020.05.458>
  - 19. Sahu K, Tiwari S, Mandal S (2020) A review on leaf disease detection using image processing. Int Res J Eng Technol (IRJET) 7(5)
  - 20. Anusha A, Divyashree B, Impana, Nisarga S (2020) Plant disease detection using image processing. Int J Eng Res Technol (IJERT), NCCDS—2020 Conference Proceedings Special issue-(2020)
  - 21. Soujanya K, Jabez J (2021) Recognition of plant diseases by leaf image classification based on improved AlexNet. In: 2nd International conference on smart electronics and communication (ICOSEC), pp 1306–1313. <https://doi.org/10.1109/ICOSEC51865.2021.9591809>
  - 22. Yadav A, Yadav PK, Toomula S, Jaiswal S, Patro P (2021) Qualitative texture analysis on detection of plant disease. In: 5th International conference on electronics, communication and aerospace technology (ICECA), pp 1758–1761. <https://doi.org/10.1109/ICECA52323.2021.9676026>
  - 23. Amsavalli Y, Mayurappriyan PS, Mohan MS (2021) Plant disease detection robot. In: International conference on advancements in electrical, electronics, communication, computing and automation (ICAEECA), pp 1–5. <https://doi.org/10.1109/ICAEECA52838.2021.9675776>

# Label Consistency-based Modified Sequential Dictionary Learning-based Approach for PIR Sensor-based Detection of Human Movement Direction



Pubali De, Amitava Chatterjee, and Anjan Rakshit

**Abstract** In this work, we intend to implement our recently proposed label consistency (LC)-based hybrid dictionary learning (DL) approach for human movement detection problems and investigate its performance in detail. In our earlier works, the direction of human movement detection problems was successfully solved using an indigenously developed hardware–software combined module; and in this connection, two other efficient DL algorithms were proposed using MMCP-and MRAK-SVD-based approaches. On the other hand, in our most recent work, we have demonstrated how another prominent activities of daily living (ADL) problem in an indoor environment, known as intruder detection problem, can be solved by using a hybridization of label consistency-based K-SVD (called LCK-SVD) algorithm with a modified version of consistent adaptive sequential DL (CAS-DL) algorithm (i.e., termed as LC-MCAS-DL). A comprehensive study carried out using an expanded database has revealed that the performances of both MMCP- and MRAK-SVD-based DL approaches reported earlier suffer degradation with an increase in the size of the database and, hence, dictionary. The present paper shows that, for an expanded database under consideration, LC-MCAS-DL-based dictionary learning approach can provide much robust result compared to both MMCP- and MRAK-SVD-based dictionary learning approaches. The present work has made a thorough investigation of this problem with the help of an expanded database comprising real signals acquired using our indigenously developed four passive infrared (PIR) sensor-based general-purpose ADL system hardware. Detailed experimental findings and performance analyses firmly demonstrate the supremacy of the LC-MCAS-DL approach over MMCP- and MRAK-SVD-based approaches for the detection of human movement direction problem at hand.

---

P. De (✉)

Electrical Engineering Department, Techno India Group, Kolkata 700141, India  
e-mail: [de.pubali19@gmail.com](mailto:de.pubali19@gmail.com)

A. Chatterjee · A. Rakshit

Electrical Engineering Department, Jadavpur University, Kolkata 700032, India

**Keywords** Dictionary learning (DL) · LC-based modified version of consistent adaptive sequential dictionary learning (LC-MCAS-DL) · Ambient-assisted living (AAL)

## 1 Introduction

Rapid advancements in AAL technologies have enabled the development of a number of advanced sensor-based sophisticated intelligent systems that are designed to monitor people [1], identify their activities [2], detect faults [3], and assess their well-being. PIR sensors have shown effectiveness in several research because of their reasonable prices, high precision, and confidentiality, which enable valuable applications [4, 5]. Almost all AAL technologies, e.g., those established for assisting the aging community in a smart home, distant monitoring, have been developed based on advanced sensor technology and superior signal processing and machine learning-based approach. These commonly used approaches are widely known as SVM [6], K-NN, and HMMs algorithms [7]. The researcher's attention has been directed to recent developments and the successes of sparse representation [8] and dictionary learning [9]-based approaches in various applications of AAL technology.

K-SVD, one of the best-known DL algorithms [9], and different variations of K-SVD were used in some of our previous publications [10–13] to illustrate the expediency of the dictionary learning technique for handling the problem of human behavior recognition. Having achieved success in [10], we investigated whether the dictionary learning strategy was effective for such different ADL problems, e.g., detecting human movement in specific directions [11, 12], detecting intruders in restricted environments [13], etc.

A general-purpose hardware-software system based on four PIR sensors has been developed in our laboratory and employed for the identification of different ADL problems [11–13]. It has been observed in [11] that our developed module can be efficiently employed in the detection of movement direction within an indoor environment in eight distinct orientations with a 45° spatial resolution (directions mentioned as NW to SE, SW to NE, NE to SW, E to W, S to N, N to S, SE to NW, and W to E).

In our earlier work [11], we developed a database consisting of six individuals (Database version 1 [11, 12]) and achieved 92.01% accuracy by introducing MCP-/MMCP-based approach, and the recognition accuracy gets improved by developing an another regularized K-SVD-based novel approach [12] and achieved 98.6% accuracy. But, in both of these works [11, 12], we have used a similar database consisting of six individuals. To investigate the performance of our proposed work in an expanded version, in this paper, we have developed a database of twenty-one individuals (Database version 2) for solving the problem of detection of human movement direction, and here first, we have evaluated the performance accuracy using previously developed MCP/MMCP [11] and regularized K-SVD [12]-based algorithm and it has been found that the performance gets diminished. So, still there is a scope of improvement in performance accuracy. On the other hand, in [13], our

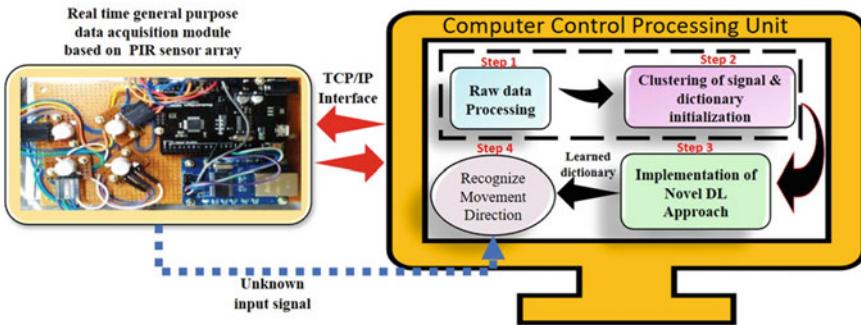
newly proposed LC-MCAS-DL algorithm outperforms other competing approaches. Encouraged by this [13], in this present work, the effectiveness of the LC-MCAS-DL technique has been examined in order to find solutions for movement detection problems.

The remaining section of this paper is arranged as follows. Section 2 presents a brief introduction to the framework for identifying the direction of human movement problems and also gives a brief explanation of our developed general-purpose hardware–software module in the laboratory. Detailed discussion of label-consistent K-SVD- [9, 14] and MCAS-DL [12, 13, 15]-based DL approaches has been provided in Sect. 3, followed by an elaborate description of our newly developed LC-MCAS-DL algorithm [13] in relation to the problem at hand. The techniques we utilized for classification in this study are described in Sect. 4. A number of real-life performance evaluations have been conducted in Sect. 5 to investigate the superiority of the LC-MCAS-DL algorithm compared to other competing approaches. The conclusion of this paper is provided in Sect. 6.

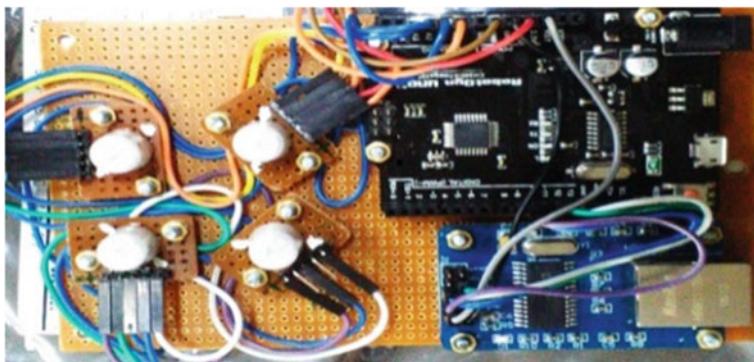
## 2 Brief Overview of Detection of Human Movement Direction Framework and General-Purpose Hardware–Software Combined IoT Module

Human activity recognition, human behavior recognition, and detection of human movement in AAL environments have recently gained significant attention [3, 16]. For elderly seniors to live independently and for remote monitoring, human movement detection is thought to be extremely significant. Here, in this context, we developed four PIR sensor-based hardware–software combined general-purpose modules that can be used to solve any kind of AAL/ADL problem and that has also been demonstrated in our recent papers [11–13].

The complete scheme used for the movement detection problem is shown in Fig. 1 using our developed real-life module. A general-purpose experimental setup [11, 13] with four PIR sensors was indigenously built in our laboratory, Jadavpur University, as shown in the real-time data collecting module in Fig. 1. As illustrated in [11], this experimental setup makes use of four PIR sensors to detect human movement in a hallway of a certain size in eight separate directions: NW, NE, SW, SE, W, E, N, and S. Four Panasonic AMN21112 analog PIR sensors have been employed in the development of this experimental device, which was also combined with a locally developed signal accumulating module using TCP/IP protocol. At the remote PC end, a software module based on the concept of GUI has been developed using visual basic and MATLAB that interfaces with the hardware as shown in Fig. 1. The dimensions of the region involved in this experiment and the field of view of each PIR sensor were deployed identically as demonstrated in [11, 12]. As depicted in Fig. 2, the PIR sensors are attached to a Robodyn Uno board, which interacts with the PC over the Ethernet LAN network module.



**Fig. 1** Overall approach to determine the direction of human movement



**Fig. 2** An illustration of the indigenously developed experimental setup [11]

The computer's control processing unit in Fig. 1 consists of four fundamental stages which reflect the basic framework for movement detection purposes. Step 1 applies FFT to the raw data obtained from the four PIR sensors in order to create feature vectors. After that, clustering has been used to build an initial dictionary from the featured signal database as shown in Stage 2. After creating an expanded signal database  $\mathbf{Y}$ , various well-known DL strategies are studied to train dictionary  $\mathbf{D}$  in smaller sizes from  $\mathbf{Y}$ . These techniques include regularized versions of DL [12], MCP-/MMCP-based DL [11], and also the performance of the recently suggested LC-MCAS-DL [13] has been examined for a different type of AAL activity, such as the detection of human movement direction as illustrated in Stage 3. Stage 4 leverages the SRC-based classification approach to instantly identify any unknown movement direction using the trained dictionary  $\mathbf{D}$ .

### 3 LC-Based DL Approach in Combination with Recently Proposed MCAS-DL

In this study, we explore the use of a newly developed LC-based DL [13, 14] method for detecting human movement, more specifically eight directions after the online data collection from four PIR sensors. In this work, we have used an expanded database, and it has been observed in the case of a larger signal database that the performance of MCP/MMCP [11, 12] and MRK-SVD/MRAK-SVD [12] gets diminished, leaving room for improvement in performance accuracy. In this context, our recently proposed LC-MCAS-DL, developed for another genre of ADL problem, has been employed in this work to investigate the performance and compared with our earlier developed other well-known DL algorithms which have already shown their effectiveness for human movement detection purposes.

#### 3.1 LC-Based DL Algorithm

LCK-SVD, the supervised variant of conventional K-SVD [13, 14], enhances the dictionary by including reconstruction and discrimination features using class information from each training signal. Through solving the objective function Eq. (1), K-SVD learns a dictionary from a more extensive signal database.

$$\min_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 \text{ s.t. } \forall i, \|\mathbf{x}_i\|_p \leq l \quad (1)$$

The training signal database  $\mathbf{Y} \in \Re^{n \times N}$  comprises  $N$  no. of  $n$ -dimensional signals. Dictionary  $\mathbf{D}$  is basically  $\mathbf{D} \in \Re^{n \times K}$  that comprises  $K$  no. of  $n$ -dimensional dictionary atoms. The sparse representation matrix is denoted as  $\mathbf{X} \in \Re^{K \times N}$ . According to K-SVD, D and X are obtained by minimizing the reconstruction error of Eq. (1).

According to [13, 14], discriminative sparse code error was utilized to guarantee label consistency, that is, equivalent sparse code representations will always occur between signals in the same class. A classification error was also incorporated into the objective function. In LCK-SVD [13, 14], therefore, the objective function is as follows:

$$\begin{aligned} & \min_{\mathbf{D}, \mathbf{W}, \mathbf{A}, \mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \alpha \|\mathbf{Q} - \mathbf{AX}\|_F^2 + \beta \|\mathbf{H} - \mathbf{WX}\|_F^2 \\ & \text{s.t. } \forall i, \|\mathbf{x}_i\|_p \leq l \end{aligned} \quad (2)$$

As discussed in [13, 14], the classification error has been incorporated in Eq. (2) by  $\|\mathbf{H} - \mathbf{WX}\|_F^2$  in which  $\mathbf{H} \in \Re^{m \times N}$  has been introduced to present class label of each input signal for learning  $\mathbf{W} \in \Re^{m \times K}$  as linear classifier. The discriminative sparse code error is given by  $\|\mathbf{Q} - \mathbf{AX}\|_F^2$  in which sparse codes with discriminative features are presented as  $\mathbf{Q} = [\mathbf{q}_1 \dots \mathbf{q}_N] \in \Re^{K \times N}$  of training database  $\mathbf{Y}$ . Here, the linear transformation matrix is present as  $\mathbf{A}$ .  $\alpha$  and  $\beta$  represent regularization parameters that control the error for discriminative factor and classification factor as given in Eq. (2). The traditional K-SVD approach is utilized in LCK-SVD for finding the trained dictionary  $\mathbf{D}$  by resolving Eq. (2). The reformulated version of Eq. (2) is given below by Eq. (3).

$$\min_{\mathbf{D}_{lc}, \mathbf{X}} \|\mathbf{Y}_{lc} - \mathbf{D}_{lc} \mathbf{X}\|_F^2 \text{ s.t. } \forall i, \|\mathbf{x}_i\|_0 \leq 1 \quad (3)$$

$$\text{where } \mathbf{Y}_{lc} = [\mathbf{Y} \ \sqrt{\alpha} \mathbf{Q} \ \sqrt{\beta} \mathbf{H}]' \quad \mathbf{D}_{lc} = [\mathbf{Y} \ \sqrt{\alpha} \mathbf{A} \ \sqrt{\beta} \mathbf{W}]'.$$

In place of K-SVD, here a newly proposed approach, i.e., MCAS-DL [13], has been adopted to solve Eq. (3) in a different way.

### 3.2 MCAS-DL Algorithm and MCAS-DL-Based LCK-SVD Algorithm

As discussed in [12, 13, 15], the basic concept of MCAS-DL is based upon an  $l_1$  norm-based dictionary learning algorithm where the appropriate choice of regularization term controls sparsity. In the case of CAS-DL [13, 15], an automatic correction has been made by applying adaptive penalization to each component of the sparse matrix. According to CAS-DL, objective function is defined as [13, 15]:

$$\min_{\mathbf{D} \in \mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + n \sum_{k=1}^K \sum_{i=1}^N \zeta_{ki} |\mathbf{x}_{ki}|, \text{ s.t. } \|\mathbf{d}_j\|_2 = 1, \forall j \quad (4)$$

In contrast to the conventional  $l_1$  norm-based dictionary learning algorithm, Eq. (4) implies that individual element gets different value of regularization parameter. Equation (5) applies the penalization to update the atom  $\mathbf{d}_j$  with the sparse code  $\mathbf{x}_T^j$  at each stage of iteration.

$$(\mathbf{d}_j, \mathbf{x}_T^j) = \arg \min_{\mathbf{d}_j, \mathbf{x}_T^j} \left\| \mathbf{E}_j - \mathbf{d}_j \mathbf{x}_T^j \right\|_F^2 + n \sum_{i=1}^N \zeta_{ji} |\mathbf{x}_{ji}| \quad (5)$$

$$\text{s.t. } \|\mathbf{d}_j\|_2 = 1$$

$$\mathbf{E}_j = \mathbf{Y} - \sum_{\substack{k=1, \\ k \neq j}}^K \mathbf{d}_k \mathbf{x}_T^k \quad (6)$$

CAS-DL technique combines the sparse coding and dictionary updating stage by using a penalized rank-1 matrix approximation.  $\mathbf{d}_j$  and  $\mathbf{x}_T^j$  could be updated by minimizing Eq. (6). By minimizing Eq. (5) and maintaining  $\mathbf{d}_j$  fixed,  $\mathbf{x}_T^j$  has been determined from Eq. (5) in the form of Eq. (7).

$$\mathbf{x}_T^j = \text{sgn}(\mathbf{d}'_j \mathbf{E}_j) \circ (|\mathbf{d}'_j \mathbf{E}_j| - \frac{n\zeta_j}{2}) \quad (7)$$

Here, sparsity of  $\mathbf{X}_T^j$  is being controlled by  $\zeta_j$  that has been incorporated to assign varying threshold for each element of  $\mathbf{X}_T^j$  without utilizing the similar weightage values for various entries of  $\mathbf{X}_T^j$ . Each entry of  $\mathbf{X}_T^j$  gets emphasized by adopting this thresholding idea. Equation (8) has been presented to promote the sparsity of  $\mathbf{X}_T^j$  in order to address the issue of concurrently adjusting all the elements of a sparse vector.

$$\zeta_{ji} = \frac{\zeta}{|\mathbf{x}_T^{ji}|^\gamma}, \gamma > 0 \quad (8)$$

A change has been included in our recent work [13] that is given in Eq. (9) to the selection phase of the vector of tuning parameter  $\zeta_j$  to make sure that the value of the sparse entries diminishes exponentially. This is done to provide each element of the sparse vector a unique weight, so that significant entries have the highest precedence over others.

$$\zeta_{ji} = e^{-\zeta |\mathbf{x}_T^{ji}|^\gamma}, \gamma > 0 \quad (9)$$

---

**Algorithm 1: LC-MCAS-DL Algorithm [16]**


---

**BEGIN**

**Input:**  $\mathbf{Y} \in \mathbb{R}^{n \times N}$  as training dataset of movement detection based on Database version 2.

**Output:** trained dictionary  $\tilde{\mathbf{D}} \in \mathbb{R}^{n \times K}$  and  $\mathbf{X}$  as coefficient matrix.

**Stage 1:****Initialization:**

- Set  $\zeta_{LC}$  a regulation parameter of sparse coefficient matrix, initialize  $\gamma > 0$
- Set  $iter_m$  as upper limit of iteration and set initial  $iter = 1$
- Initialize  $\mathbf{Q}$  i.e., discriminative sparse code and  $\mathbf{H}$  i.e. class labels of training signal database  $\mathbf{Y}$
- Initialize  $\alpha$  and  $\beta$  that control the influence of discriminative and sparse code error
- Initialize dictionary size as  $K$ .

**Stage 2:**

$\mathbf{D}_0 \in \mathbb{R}^{n \times K}$  is initialized by concatenating all distinct class precise dictionary determined using Eq. (1).

**Stage 3:**

Initial sparse matrix  $\mathbf{X}_0$  can be obtained by using any greedy or relaxation approach and  $\mathbf{A}_0$  and  $\mathbf{W}_0$  are initialized using multivariate regression.

**Stage 4:**

$\mathbf{Y}_{LC}$  and  $\mathbf{D}_{LC}$  are initialized following  $\mathbf{y}_{lc} = [\mathbf{y} \quad \sqrt{\alpha}\mathbf{Q} \quad \sqrt{\beta}\mathbf{H}]$  and  $\mathbf{d}_{lc} = [\mathbf{y} \quad \sqrt{\alpha}\mathbf{A} \quad \sqrt{\beta}\mathbf{W}]$  respectively.

**FOR**  $iter = 1 : iter_m$ 

- every column of  $\mathbf{D}_{LC}$  i.e.,  $\mathbf{d}_{LCj}$  is updated by resolving Eq. (11) using MCAS-DL [16].

**ENDFOR**

- $\tilde{\mathbf{D}}, \tilde{\mathbf{A}}, \tilde{\mathbf{W}}$  are calculated following

$$\tilde{\mathbf{D}} = \left\{ \frac{\mathbf{d}_j}{\|\mathbf{d}_j\|_2} \right\}, \tilde{\mathbf{A}} = \left\{ \frac{\mathbf{a}_j}{\|\mathbf{d}_j\|_2} \right\}, \tilde{\mathbf{W}} = \left\{ \frac{\mathbf{w}_j}{\|\mathbf{d}_j\|_2} \right\}, \forall j.$$

**END**


---

In accordance with our prior work [13], the notion of penalization has been combined with the traditional LCK-SVD to add penalty in individual element of the sparse code matrix  $\mathbf{X}$  for tuning sparsity in an adaptive manner. In this context, Eq. (2) has been integrated with Eq. (4) and Eq. (10) has been developed. Reformulated version of Eq. (10) is given below by Eq. (11).

$$\min_{\mathbf{D}, \mathbf{W}, \mathbf{A}, \mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \alpha \|\mathbf{Q} - \mathbf{AX}\|_F^2 + \beta \|\mathbf{H} - \mathbf{WX}\|_F^2 + n \sum_{k=1}^K \sum_{i=1}^N \zeta_{ki} |\mathbf{x}_{ki}| \\ s.t. \|\mathbf{d}_j\|_2 = 1, \forall j \quad (10)$$

$$\min_{\mathbf{D}_{LC}, \mathbf{X}} \|\mathbf{Y}_{LC} - \mathbf{D}_{LC}\mathbf{X}\|_F^2 + n \sum_{k=1}^K \sum_{i=1}^N \zeta_{LCki} |\mathbf{x}_{ki}| s.t. \|\mathbf{d}_{LCj}\|_2 = 1, \forall j \quad (11)$$

and  $\mathbf{Y}_{LC} = [\mathbf{Y} \sqrt{\alpha} \mathbf{Q} \sqrt{\beta} \mathbf{H}]'$ ,  $\mathbf{D}_{LC} = [\mathbf{Y} \sqrt{\alpha} \mathbf{A} \sqrt{\beta} \mathbf{W}]'$ .

In this paper, that novel version of  $\zeta_{LCj}$  [13] is adopted to compute the superior version of sparse matrix  $\mathbf{X}$  [13] as shown in Eq. (12).

$$\zeta_{LCji} = e^{-\zeta_{LC} |\mathbf{x}_T^{ji}|^\gamma}, \gamma > 0 \quad (12)$$

Dictionary initialization has been done here following the similar concept proposed in [13].  $\mathbf{A}$  and  $\mathbf{W}$  of Eq. (10) are initialized using the multivariate regression model, as mentioned in [13]. Algorithm 1 describes LC-MCAS-DL in a nutshell.

## 4 Classification Approach Based on LC-MCAS-DL

Once dictionary  $\mathbf{D}$  has been learnt, Fig. 1 demonstrates how to solve an identification problem by successfully solving the optimization problem mentioned in Eq. (13) [13, 14]. This method identifies an unknown signal as a human movement in a certain direction.

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_p \text{s.t. } \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \tau \quad (13)$$

$\mathbf{D}$ ,  $\mathbf{A}$ , and  $\mathbf{W}$  have been derived from  $\mathbf{D}_{LC}$  by using hybrid dictionary learning approach where we obtain  $\mathbf{D}$ ,  $\mathbf{A}$ , and  $\mathbf{W}$  in a normalized form of  $\mathbf{D}_{LC}$ . Hence, we cannot resolve the sparse code by using the dictionary  $\mathbf{D}$  in a straight way and has been calculated as mentioned in Algorithm 1. In this context, the sparse code  $\hat{\mathbf{x}}$  is computed by solving Eq. (13) for the unknown signal being considered, and then, using a linear predictive classifier,  $\mathbf{c} = \hat{\mathbf{W}}\hat{\mathbf{x}}$  can be computed. This will allow us to determine human motion detection problem as a multiclass classification problem or in this case SRC-based classification procedure can be used [12].

## 5 Performance Evaluation

According to [11–13], a specific area in the ground has been used within our laboratory for motion detection purposes. Our signal database has been formed in a similar manner as in [11] but, this time, by considering many more volunteers and, hence, utilizing a much expanded database. Here, we have collected real-life signals from twenty-one volunteers who have heights in the range of 1.58–1.82 m, weights in the range of 52–82 kg, and different walking patterns and walking in eight directions. After that, in a real-life scenario, the entire motion detection system has been executed in an online fashion. In this context, data have been gathered based on 30 sample signals per direction per person, and hence, the total accumulated signal database comprises 5040 movement samples.

In our previous work [11, 12], we have solved motion detection problem based on a database which consists of six subjects (Database version 1: 1440 samples), and in this work, we have developed an expanded database of twenty-one subjects (Database version 2: 5040 samples). Following our earlier research [11–13], which intended to increase the over completeness of the dictionary to maximize performance accuracy, size of each signal has been reduced by performing the FFT on each dictionary atom in similar manner as discussed in [12][12]. After executing the feature extraction, we obtain the signal database with the size of  $[24 \times 5040]$ . In this case, 80% of the signal database was utilized for training and the remaining 20% for testing. After creating signal database  $\mathbf{Y}$ , next, we also have clustering to initialize the dictionary as mentioned in [20], and then, it performs the dictionary learning step.

Next, we have evaluated the performance of our earlier proposed MCP-based K-SVD/MMCP [11]-based K-SVD/MRK-SVD/ MRAK-SVD [12] algorithm for motion detection purpose based on Database version 2, and results have been compared with Database version 1 [11].

It has been found from Table 1 that the performance accuracy for this Database version 2 gets diminished compared to Database version 1 and MRAK-SVD emerges as the best version among four commonly known competent algorithms for human motion detection purpose. Keeping this in mind, we first examine to investigate how well LCK-SVD [13, 14] performs for various  $\alpha$  and  $\beta$  variants based on Database version 2 when learning the dictionary  $D$ .

Table 2 summarizes the performance measure of the LCK-SVD in handling the human motion detection problem for Database version 2. The study of Table 2 shows that the optimal combination, where LCK-SVD reaches 91.07%, is a selection of  $\alpha = 20$  and  $\beta = 5$ . Keeping values of  $\alpha$  and  $\beta$  constant, for improving the performance, we have implemented the concept of label consistency-based MCAS-DL [13], and next, we have investigated the performance of MCAS-DL [13] and LC-MCAS-DL [13] for Database version 2 elaborately and reported the results obtained in Table 3. When an expanded database is considered, LC-MCAS-DL appeared to be as a superior approach for detecting human movement direction.

It has been determined that recognition accuracy 94.04% has been obtained when values of  $\gamma$ ,  $\beta$ ,  $\zeta$ , and  $\alpha$  appeared as 6, 5, 20, and 20, correspondingly. According

**Table 1** Performance comparison of well-known algorithms of human movement direction detection purposes for Database version 2 and Database version 1

Method	Performance accuracy (%)	
	Database version 2: 5040 samples	Database version 1: 1440 samples [11]
MCP-based K-SVD [11]	88.29	90.60
MMCP-based K-SVD [11]	90.27	92.01
MRK-SVD [12]	92.46	95.83
MRAK-SVD [12]	<b>92.65</b>	<b>98.60</b>
CAS-DL [15, 12]	80.35	71.90

**Table 2** Analysis of LCK-SVD performance for Database version 2 for variation of  $\alpha$  and  $\beta$ 

$\beta \downarrow \alpha \rightarrow$	Performance accuracy (%)					
	1	2	5	10	20	30
1	66.68	67.07	67.47	75.9	74.39	65.94
5	<b>70.65</b>	<b>75.41</b>	80.34	<b>80.32</b>	<b>91.07</b>	74.09
10	70.25	71.34	<b>81.14</b>	79.86	77.36	73.69
20	70.05	70.84	66.57	76.09	68.4	71.51

**Table 3** Performance comparison of MCAS-DL and LC-MCAS-DL algorithm for Database version 2 for different variant of hyperparameter

Initial value of $\zeta$	Performance accuracy (%)					
	$\gamma = 2$		$\gamma = 3$		$\gamma = 6$	
	MCAS-DL [13]	LC-MCAS-DL [13]	MCAS-DL [13]	LC-MCAS-DL [13]	MCAS-DL [13]	LC-MCAS-DL [13]
5	87.10	88.29	87.30	86.30	86.51	79.06
10	87.60	69.44	<b>88.10</b>	90.77	85.71	91.26
20	<b>88.89</b>	<b>90.27</b>	87.90	<b>92.85</b>	<b>87.80</b>	<b>94.04</b>

to the results reported in Tables 1 and 2, for the expanded Database version 2, LC-MCAS-DL achieved a recognition accuracy of 94.04% which makes it the most efficient alternative compared to recently proposed intelligent algorithms that could reach a maximum accuracy of 92.65%.

Tables 4(a-c) present the confusion matrices in which overall recognition accuracies for all eight directions have been presented for Database version 2, for three emerging DL-based competing algorithms considered in Tables 2 and 3. The LC-MCAS-DL method, which achieves 100% accuracy in four out of eight classes, shows the highest degree of recognition ability in detecting all individual directions of interest. On the other hand, LCK-SVD and MCAS-DL both could achieve 100%

accuracy in two out of eight classes only. A comparison of the LC-MCAS-DL algorithm with the LCK-SVD and MCAS-DL algorithms shows that the former has either superior or similar accuracy, depending on the direction.

**Table 4** Confusion matrices for **a** LCK-SVD, **b** MCAS-DL, and **c** LC-MCAS-DL

P → A ↓	D1	D2	D3	D4	D5	D6	D7	D8
<b>D1(EW)</b>	<b>100</b>	0	0	0	0	0	0	0
<b>D2(NEW)</b>	0	<b>86.51</b>	5.5	7.14	0	0	0	0
<b>D3(NS)</b>	0	0	<b>100</b>	0	0	0	0	0
<b>D4(NWSE)</b>	0	0	8.73	<b>80.16</b>	1.53	0	0	7.93
<b>D5(SENW)</b>	5.5	0	0	6.3	<b>79.37</b>	11.1	2.8	0
<b>D6(SN)</b>	0	0	0	1.58	0	<b>95.24</b>	3.17	0
<b>D7(SWNE)</b>	0	0	0	3.96	0	0	<b>92.06</b>	3.96
<b>D8(WE)</b>	0	0	0	0	4.76	0	0	<b>95.24</b>

(a)

P → A ↓	D1	D2	D3	D4	D5	D6	D7	D8
<b>D1(EW)</b>	<b>100</b>	0	0	0	0	0	0	0
<b>D2(NESW)</b>	2.38	<b>84.92</b>	7.9	4.76	0	0	0	0
<b>D3(NS)</b>	0	0	<b>100</b>	0	0	0	0	0
<b>D4(NWSE)</b>	0	0	8.7	<b>76.19</b>	7.14	0	0	7.14
<b>D5(SENW)</b>	7.93	0	0	0	<b>79.37</b>	7.93	4.76	0
<b>D6(SN)</b>	0	0	0	0	0	<b>91.27</b>	8.73	0
<b>D7(SWNE)</b>	0	3.96	0	0	0	0	<b>92.06</b>	3.96
<b>D8(WE)</b>	7.93	0	0	0	0	0	4.76	<b>87.30</b>

(b)

P → A ↓	D1	D2	D3	D4	D5	D6	D7	D8
<b>D1(EW)</b>	<b>100</b>	0	0	0	0	0	0	0
<b>D2(NESW)</b>	7.1	<b>87.30</b>	5.6	0	0	0	0	0
<b>D3(NS)</b>	0	0	<b>100</b>	0	0	0	0	0
<b>D4(NWSE)</b>	0	0	6.3	<b>84.13</b>	0	5.5	0	3.96
<b>D5(SENW)</b>	0	0	0	0	<b>85.71</b>	7.93	6.3	0
<b>D6(SN)</b>	0	0	0	0	0	<b>100</b>	0	0
<b>D7(SWNE)</b>	0	0	0	0	0	0	<b>95.24</b>	4.76
<b>D8(WE)</b>	0	0	0	0	0	0	0	<b>100</b>

(c)

## 6 Conclusion

This study evaluates the performance of our newly suggested LC-MCAS-DL algorithm [13] that integrates the ideas of label consistency with an MCAS-DL method, when combined with PIR sensor-based hardware configuration to create a human motion detection system for an enlarged database (Database version 2:5040 samples). By integrating the advantageous features of the LCK-SVD and MCAS-DL algorithms, the LC-MCAS-DL approach was created, and it has been established from the investigation that the performance of the motion detection problem gets improved for LC-MCAS-DL compared to earlier proposed state-of-the-art approaches.

## References

1. Hao Q, Brady DJ, Guenther BD et al (2006) Human tracking with wireless distributed pyroelectric sensors. *IEEE Sens J* 6:1683–1696
2. Andò B, Baglio S, Lombardo CO, Marletta V (2015) An event polarized paradigm for ADL detection in AAL context. *IEEE Trans Instrum Meas* 64(7):1814–1825
3. Yazar A, Keskin F, Töreyin BU, Çetin AE (2013) Fall detection using single-tree complex wavelet transform. *Pattern Recogn Lett* 34(15):1945–1952
4. Yun J, Song MH (2014) Detecting direction of movement using pyroelectric infrared sensors. *IEEE Sens J* 14(5):1482–1489
5. Jin X, Sarkar S, Ray A, Gupta S, Damarla T (2012) Target detection and classification using seismic and PIR sensors. *IEEE Sens J* 12(6):1709–1718
6. Yu J (2012) Health condition monitoring of machines based on hidden Markov model and contribution analysis. *IEEE Trans Instrum Meas* 61(8):2200–2211
7. Dardas NH, Georganas ND (2011) Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Trans Instrum Meas* 60(11):3592–3607
8. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2009) Robust face recognition via sparse representation. *IEEE Trans Pattern Anal Mach Intell* 31(2):210–227
9. Aharon M, Elad M, Bruckstein A (2006) The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans Signal Process* 54(11):4311–4322
10. De P, Chatterjee A, Rakshit A (2018) Recognition of human behavior for assisted living using dictionary learning approach. *IEEE Sensors J* 18(6):2434–2441
11. De P, Chatterjee A, Rakshit A (2020) PIR sensor based AAL tool for human movement detection: modified MCP based dictionary learning approach. *IEEE Trans Instrum Meas* 69(10):7377–7385
12. De P, Chatterjee A, Rakshit A (2021) Regularized K-SVD based dictionary learning approaches for PIR sensor-based detection of human movement direction. *IEEE Sens J* 21(5):6459–6467
13. De P, Chatterjee A, Rakshit A (2022) PIR sensor based surveillance tool for intruder detection in secured environment: a label consistency based modified sequential dictionary learning approach. *IEEE Internet Things J* 9(20):20458–20466
14. Jiang ZL, Lin Z, Davis LS (2013) Label consistent K-SVD: learning a discriminative dictionary for recognition. *IEEE Trans Pattern Anal Mach Intell* 35(11):2651–2664
15. Seghouane AK, Iqbal A (2018) Consistent adaptive sequential dictionary learning. *Signal Process* 153:300–310
16. Debes C, Merentitis A, Sukhanov S et al (2016) Monitoring activities of daily living in smart homes: understanding human behavior. *IEEE Signal Process Mag* 33:81–94

# Analyzing Lung Diseases Using CNN from Chest X-ray Images



Sanhita Dan, Arpan Garai, and Samit Biswas

**Abstract** As recent literature suggests, one of the most reliable way for predicting lung diseases is chest X-ray imaging. Numerous works on binary classification of COVID-19 affected from chest X-rays (CXR) using deep learning (DL) techniques have been found in the literature. However, the study on multiclass lung disease detection is yet quite limited. To fill this gap, this research aims to fine-tune three CNN architectures, such as ResNet-50, VGG-19, and Inception-V3 model, on the chest X-ray dataset to classify the CXRs into five categories, namely COVID-19, tuberculosis (TB), bacterial pneumonia, viral pneumonia, and healthy CXRs. Here, we have implemented different data augmentation techniques and trained our pre-processed data on the modified above-mentioned CNN models. The performance of the fine-tuned CNNs with the pre-trained models are also compared.

**Keywords** Chest X-ray (CXR) · Multiclass · CNN models · Fine-tuned

## 1 Introduction

Lung disorders are illnesses that damage the lung's airways and other lung components and are frequently referred to as respiratory diseases [1]. Pneumonia, TB, and coronavirus are examples of lung diseases. According to WHO [2], TB is the world's 13th leading deadliest disease and, after COVID-19, the second fatal viral killer. At the same time, millions of people die from pneumonia. Hence, early identification of these lung diseases and planning preventative measures are essential. As suggested in [3], skin tests, sputum tests, blood sample tests, chest X-ray (CXR) examinations, and CT scans have been used previously to identify lung diseases. RT-PCR test is

---

S. Dan (✉) · S. Biswas

Department of CST, IIEST, Shibpur, Howrah, West Bengal, India  
e-mail: [sanhitadan15@gmail.com](mailto:sanhitadan15@gmail.com)

S. Biswas

e-mail: [samit@cs.iiests.ac.in](mailto:samit@cs.iiests.ac.in)

A. Garai

Department of CSE, IIT, Delhi 110016, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

197

A. K. Das et al. (eds.), *Computational Intelligence in Pattern Recognition*,

Lecture Notes in Networks and Systems 725,

[https://doi.org/10.1007/978-981-99-3734-9\\_17](https://doi.org/10.1007/978-981-99-3734-9_17)

used for the identification of COVID-19 disease, but this procedure is complex and involves a significant error rate [4–6]. Detecting lung diseases using CT images has some drawbacks, as its screening expenses are high. On the contrary, CXR screening is affordable and readily available in various hospitals and radiology centers [7]. However, because the CXRs of other pneumonias (bacterial, viral) and COVID-19 are quite similar, manual differentiation of these lung anomalies is challenging. However, deep learning algorithms can extract a variety of image-based features from the original CXR that radiologists might not be able to see visually. Existing lung disease detection methods are mostly based on the binary identification of COVID-19 CXR and non-COVID-19 CXRs.

This paper aims to classify efficiently different lung diseases (bacterial pneumonia, viral pneumonia, COVID-19, tuberculosis, and healthy CXR) using CNN-based methods. As the recent literature suggests, Inception-V3, VGG-19, and ResNet-50 have performed with a significant accuracy in similar challenges. So, in the proposed work, these three networks are studied for this above-mentioned multiclass classification. We have collected different datasets, used dataset preprocessing techniques, data augmentation, and disease anticipation and provided the comparative analysis of the above-mentioned deep learning methods.

The contributions of this research work are listed as follows:-

1. In lieu of binary classification, we have considered multiclass classification of lung-related disorders, and to the best of our knowledge, we have first considered the issue of classification of these specific five classes (as shown in Fig. 1) conjointly.
2. In order to fine-tune our approached CNN models and for maximizing the performance of the models, we have added 1 flattening layer, 1 batch normalization, 2 dropout layers, and 4 dense layers.

The remaining paper is constructed as follows:- Section 2 describes the related work. Next, the dataset used and the proposed methodology are explained in Sect. 3. The experimental results and the evaluation of the modified approaches and the pre-trained models are provided in Sect. 4. Finally, Sect. 5 draws the conclusion of the research work.

## 2 Related Work

As mentioned earlier, a number of methods are already been proposed for lung disease classification from CXRs. Some of the recent approaches are discussed in this section. Mostofa et al. [8] have used VGG-16 to identify tuberculosis from healthy groups using CXR images. They employed 1324 CXRs as input to their model, which obtained 80.0% accuracy.

Xianghong Gu et al. [9] have suggested a method to differentiate pneumonia bacterial from pneumonia viral in children using CXR samples and using pre-trained DCNN model. The authors claimed to achieve an accuracy of 80.48%.

Colombo et al. [10] have investigated the diagnosis of pulmonary tuberculosis using GoogleNet, AlexNet, and ResNet. GoogleNet obtained a maximum accuracy of 75% on the the Montgomery, Shenzhen, and PadChest datasets.

Shelke et al. [11] have merged three CNN models. To categorize pneumonia and TB using CXR images as input, VGG-16 architecture has been used. For further classification of the previously categorized pneumonia samples into pneumonia and COVID-19, DenseNet has been used. Next, the COVID samples have been passed into ResNet18 to categorize them as mild, moderate, or serious. The DenseNet, VGG, and ResNet models had an accuracy of 98.9%, 95.9%, and 87.76%, respectively.

Using 400 CXR and 400 CT samples, Ashan et al. [12] have performed a study to identify COVID-19 patients from the CT samples and CXR samples, using six CNN models, namely VGG-16, MobileNet-V2, InceptionResNet-V2, ResNet-50, ResNet-101, and VGG-19. MobileNet-V2 outperforms all other models, with 82.94% accuracy on a dataset with CXRs and 93.94% on a dataset with CT samples.

The EfficientNet-B5 model has been improved by Chetoui et al. [13] for both binary classification and multiclass classification. DeepCCXR-Bin has been used for the classification of COVID-19, and no findings and DeepCCXR-Multi have been used for multiclass classification of Covid-19 CXRs, Pneumonia CXRs, and CXRs of no findings. Additionally, testing has been performed on nine datasets individually, producing an accuracy ranging from 91% to 98% for DeepCCXR-Bin and 70% to 93% for DeepCCXR-Multi.

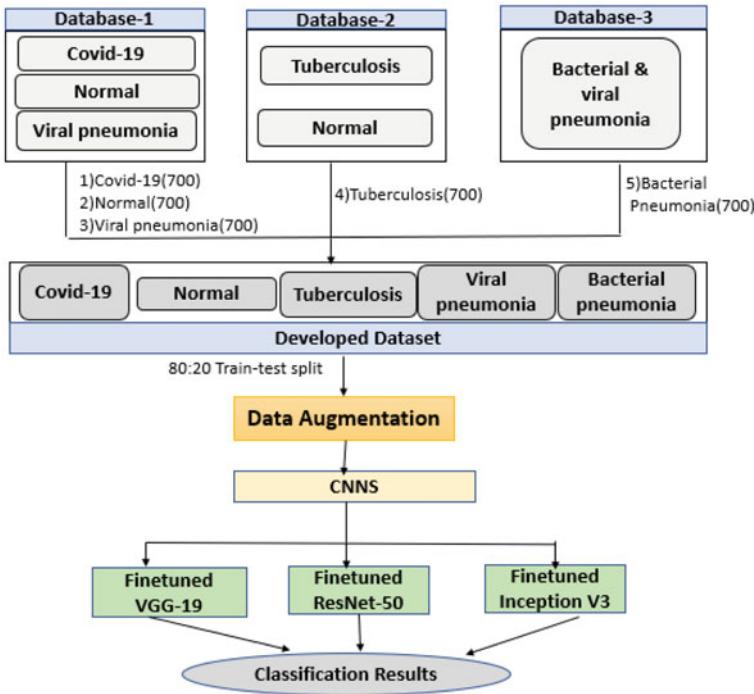
Kim et al. [14] have classified the NIH dataset into the following categories: normal, pneumothorax, and pneumonia using EfficientNet-v2M. The model has achieved a validation loss of 0.6933 and accuracy of 82.15

An attention-based technique with VGG-16 has been used to identify 101 CXRs with the help of transfer learning by Sitala et al. [15]. The authors have classified the CXR samples into five classes: normal, COVID-19, no results, bacteria pneumonia, and viral pneumonia with the highest accuracy of 87.49%.

Since most of the research works are about the binary identification of COVID-19 CXRs vs non-COVID CXRs or viral vs bacterial pneumonia or tuberculosis vs healthy CXRs, so we have proposed a methodology for the multiclass classification of five different lung-related (as mentioned in Fig. 1) diseases combinedly.

### 3 Materials and Methods

The approach has started with the image of chest X-Ray. In the following subsections, we have discussed how the dataset is built, different data augmentation techniques, and the architecture and implementation of our proposed models. The overview of the approach is depicted in Fig. 1.



**Fig. 1** Overview of the proposed multiclass lung disease detection methodology

### 3.1 Development of Dataset

To the best of our knowledge, there is no such dataset that contains all five different classes together as mentioned earlier. So, a combined form of the dataset is prepared from three publicly available datasets from Kaggle. They are (1) COVID-19 radiography database [16], (2) TB chest X-ray database [17], (3) chest X-ray images (pneumonia) [18]. The database (COVID-19 radiography database) contains 3616 CXRs of COVID-19, 10, 192 CXRs of normal, and 1345 CXRs of viral pneumonia patients. The samples in the dataset have a resolution of  $299 \times 299$  pixels. Tuberculosis chest X-ray database [17] has 700 TB CXR samples and 3500 normal CXR images. The images in the dataset are of  $512 \times 512$  pixel size. Chest X-ray images (pneumonia) database [18] comprises of 5860 CXR samples in total, among which 2780 samples are of bacterial pneumonia, and the remaining are of viral pneumonia. The CXRs are in different resolutions. We have created image sub-folders based on the image labels to separate the bacterial pneumonia images from viral pneumonia images. To handle the data imbalance of different classes, the sample length of each of the five classes is set to 700, as the least length of the available classes is 700 (tuberculosis). Hence, the final dataset for our work contains 3500 CXRs in total. All the CXR images have been re-scaled to  $224 \times 224$  pixels, and then these values

have been normalized from 0 to 255 to between 0 and 1. In this work, we have used an 80:20 dataset split for training and testing/validation purposes for our fine-tunned CNN models. The train data shape consists of 2800 CXR images (560 samples from each of the five classes), and the test and validation data shape consists of 700 CXRs (140 samples from each of the five classes).

### ***3.2 Techniques for Data Augmentation***

Data augmentation operation is used for overcoming the problem of preparing big datasets and overfitting issues brought on by data scarcity.

In this research work, since the number of samples to train the models are not sufficient (2800 samples), hence to expand the amount of data by the addition of marginally modified copies of existing samples, we have used different image transformation techniques for our CNN models. The CXRs are randomly rotated to the limit of  $\pm 15$  degrees and are flipped horizontally, the shear range and zoom range are arbitrarily adjusted to the limit of 0.2, and the fill mode is set to near.

### ***3.3 Implementation of Different Fine-Tuned CNN-Based Architectures***

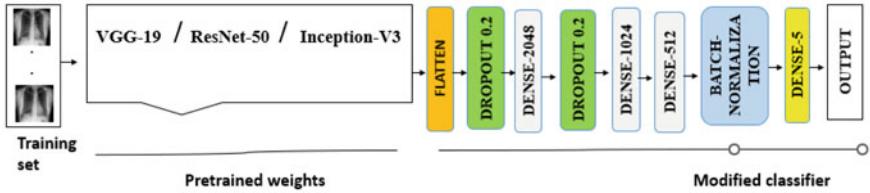
For the classification challenge, we have considered three widely used pre-trained models such as VGG-19 [19], ResNet-50 [20], and Inception-V3 [21]. Next, these networks are fine-tuned and denoted as **VGG-19+**, **ResNet-50+**, and **Inception-V3+**, respectively, throughout this paper. Using transfer learning, the existing convolution layers have been merged with modified fully connected layers, batch normalization, and dropout layers.

#### **3.3.1 Input Layer**

The input images have been normalized to the pixel size of  $224 \times 224$  before being provided as input to the freezed convolution layers of the model.

#### **3.3.2 Pre-trained(Freezed) Convolution Layers**

In this research, we have used three popular CNN models and preserved the convolution layers to extract features from the image samples [22].



**Fig. 2** Approached CNN models architecture

### 3.3.3 Additional Layers

The architecture of our fine-tuned CNN models is shown in Fig. 2.

The output of the final pooling is passed to the flattening layer  $P_1$ , and the dropout of value 0.2 has been calculated in the dropout layer (Eq. 1). Then, the dropout layer's output has been transferred to the dense layer with an output size of  $2048 \times 2048$  (Eqs. 2, 3). Again, another dropout layer of value 0.2 is applied which is followed by another two consecutive dense layers of output size  $1024 \times 1024$  and  $512 \times 512$ , respectively (Eqs. 4, 5, 6, 7, and 8). Finally, after the batch normalization layer, the output is passed through another dense layer of size 5 which is the output layer with the softmax classifier (Eq. 9).

$$P_2 = \Sigma \Delta P_1 [\Delta = 0.2] \quad (1)$$

$$D_1 = \sum_{i=1}^{2048} \omega_i \lambda_i + \beta \quad [\omega = \text{weight}, \beta = \text{bias}, \lambda_i = P_2] \quad (2)$$

$$P_3 = \max(0, D_1) \quad (3)$$

$$P_4 = \Sigma \Delta P_3 [\Delta = 0.2] \quad (4)$$

$$D_2 = \sum_{i=1}^{1024} \omega_i \lambda_i + \beta \quad [\omega = \text{weight}, \beta = \text{bias}, \lambda_i = P_4] \quad (5)$$

$$P_5 = \max(0, D_2) \quad (6)$$

$$D_3 = \sum_{i=1}^{512} \omega_i \lambda_i + \beta \quad [\omega = \text{weight}, \beta = \text{bias}, \lambda_i = P_5] \quad (7)$$

$$P_6 = \max(0, D_3) \quad (8)$$

$$D_4 = \sum_{i=1}^5 \omega_i \lambda_i + \beta \quad [\omega = \text{weight}, \beta = \text{bias}, \lambda_i = P_6] \quad (9)$$

## Implementation

The CNN models based on transfer learning are implemented using Keras. With a learning rate of  $5e - 5$  and a batch size of 8, the models were trained using Adam as the optimizer and the categorical cross-entropy as the cost function (loss function). The steps per epoch are set to 350 (no. of samples/batch size). As a result, in every epoch, the number of augmented images is (batch size  $\times$  steps per epoch) 2800. As shown in Fig. 1, firstly, the datasets from three publicly available databases are combined and are arbitrarily divided into two subsets (80% as the training dataset and 20% as the validation as well as testing dataset), which are then fed into the ResNet-50+, Inception-V3+, and VGG-19+ architectures. The ImageDataGenerator class transforms all the 2800 training samples. In each epoch, the model will get different variations of the samples. Since our model is trained for 10 epochs, hence, in total,  $10 \times 2800$  variations of the images will be generated and fed into the training model. The model's hyperparameters are improved with the help of the validation dataset, and the test dataset is used to assess how well our models perform.

## 4 Experimental Results Analysis

In this segment, we have briefly discussed the evaluation parameters used to predict the performance of our fine-tuned models on our prepared dataset.

### 4.1 Evaluation Parameters

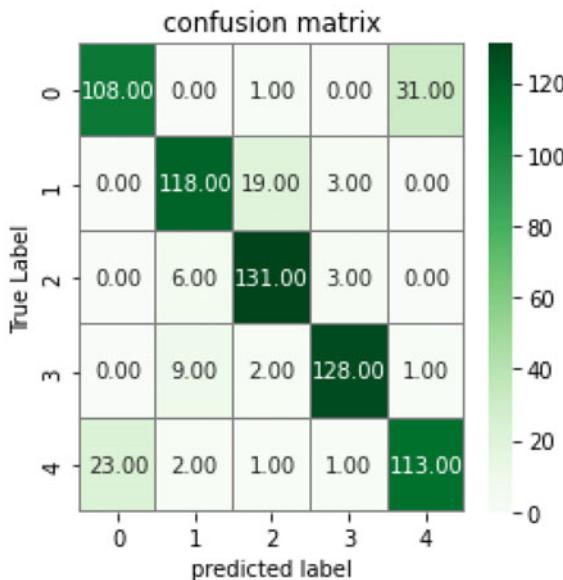
In this research work, the evaluation criteria we have used are accuracy, precision, recall, and F1-score. The proportion of correctly anticipated outcomes to all outcomes is known as the accuracy score [23]. The precision score [23] is the proportion of accurately recognized positive outcomes to all projected positive outcomes (correctly or incorrectly). The recall score [23] is the ratio of the correctly recognized positive outcomes and the total number of positive outcomes. The harmonic mean of the precision score and recall score is known as the F1-score [23].

### 4.2 Results

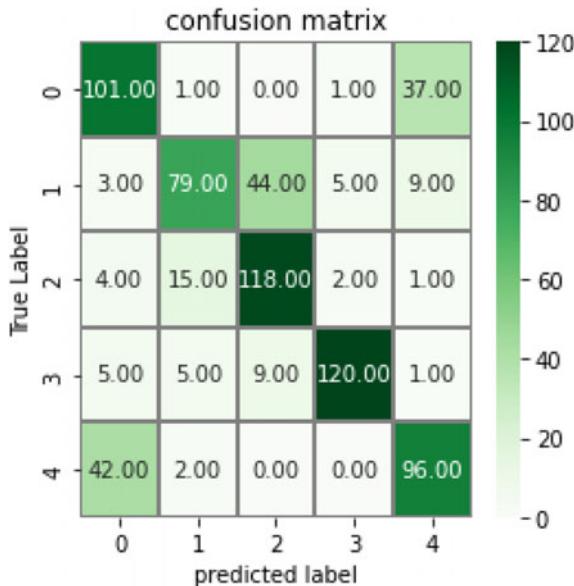
As discussed earlier, 700 CXRs from each of the five classes, i.e., a total of 3500 CXRs are considered for the investigation. For the CNN models, arbitrarily chosen

**Table 1** Performance analysis table

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
ResNet-50	51.48	51	51	50
VGG-19	73	73	73	72
Inception-V3	76.71	77	77	76
ResNet-50+	79.71	80	80	79
Inception-V3+	82.14	83	82	82
VGG-19+	85.29	85	85	85

**Fig. 3** VGG-19+

(80 : 20 split) 2800 samples (560 samples from each of the five classes) from the dataset are used for the training, and 700 samples (140 samples from each of the five classes) are used for both validation as well as test purpose. Table 1 compares the performance of the fine-tuned models with the state-of-the-art (SOTA) models. In Figs. 3, 4, and 5, we have the confusion matrices of our fine-tuned CNN networks. The labels on the left axis are the anticipated labels from our model, and the labels on the bottom axis are the actual labels [24]. From the confusion matrices, it is clear that the models other than VGG-19+ had trouble differentiating between pneumonia bacterial, pneumonia viral, and COVID-19. It is clear from Table 1 that the VGG-19+ architecture achieved the highest accuracy that is 85.29%, followed by the Inception-V3+(82.14%) and ResNet-50+(79.71%). The modified CNN model's performance is compared with the pre-trained models. From Table 1, we can see that the performance

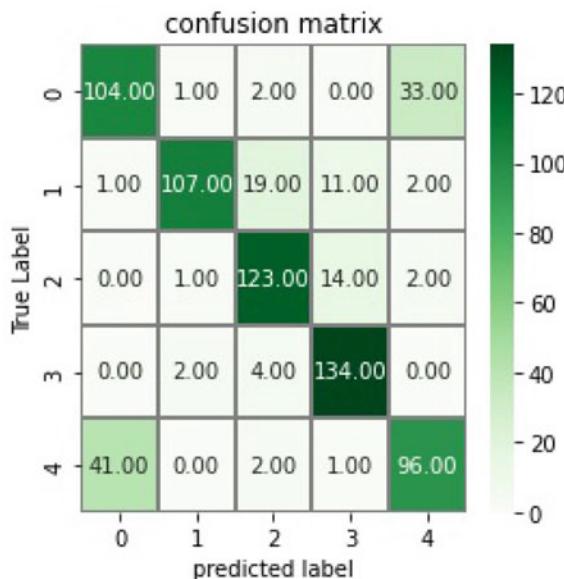


**Fig. 4** ResNet-50+

of our fine-tuned CNN models is improved by at least 8% to 28%. The pre-trained models are unable to fit the dataset correctly, and the classification accuracy is less, i.e., 51.48% for ResNet-50. Moreover, from the F1-score, precision, and recall, it is clear that most of the classes are wrongly classified, whereas our modified model performs better in terms of the above-mentioned parameters.

## 5 Conclusion

In this paper, we successfully implemented a fine-tuned VGG-19+, ResNet-50+, and Inception-V3+ architecture that can detect lung disease with significant accuracy. Nevertheless, the used dataset can be expanded with more CXRs of each class for better performance. The fine-tuned VGG-19+ model outperformed all the other used models by a healthy margin, achieving an accuracy of 85.29% and loss value of 0.37. Despite the fact that these models produced leading results in the multiclass categorization of CXRs, further advancement is possible through an improved data augmentation and/or preprocessing techniques. It is observed that the models find it difficult to classify the classes such as bacterial and viral pneumonia, possibly because of the relevant features are obscured by the less relevant parts of the image, which may lead to the future scope of this work.



**Fig. 5** Inception-V3+

## References

1. Cruz AA (2007) Global surveillance, prevention and control of chronic respiratory diseases: a comprehensive approach. World Health Organization. <https://apps.who.int/iris/handle/10665/43776>
2. <https://www.who.int/news-room/fact-sheets/detail/tuberculosis>. Accessed 15 Mar 23
3. Ferrara G et al (2005) Routine hospital use of a new commercial whole blood interferon- $\gamma$  assay for the diagnosis of tuberculosis infection. Am J Respiratory Critical Care Med 172(5):631–635
4. Wang W et al (2020) Detection of SARS-CoV-2 in different types of clinical specimens. JAMA 323(18):1843–1844
5. Wikramaratna PS et al (2020) Estimating the false-negative test probability of SARS-CoV-2 by RT-PCR. Eurosurveillance 25(50):2000568
6. Li Y et al (2020) Stability issues of RT-PCR testing of SARS-CoV-2 for hospitalized patients clinically diagnosed with COVID-19. J Med Virol 92(7):903–908
7. Pertile P et al (2015) Is chest X-ray screening for lung cancer in smokers cost-effective? Evidence from a population-based study in Italy. Cost Effect Res Allocat 13(1):1–12
8. Ahsan M, Gomes R, Denton A (2019) Application of a convolutional neural network using transfer learning for tuberculosis detection. In: 2019 IEEE international conference on electro information technology (EIT). IEEE, pp 427–433. <https://doi.org/10.1109/EIT.2019.8833768>
9. Xianghong G et al (2018) Classification of bacterial and viral childhood pneumonia using deep learning in chest radiography. In: Proceedings of the 3rd international conference on multimedia and image processing, pp 88–93. <https://doi.org/10.1145/3195588.3195597.3195597>
10. Filho C, Eloí M et al (2020) Preliminary results on pulmonary tuberculosis detection in chest x-ray using convolutional neural networks. In: Computational science-ICCS 2020: 20th international conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part IV. Springer International Publishing, Cham. [https://doi.org/10.1007/978-3-030-50423-6\\_42](https://doi.org/10.1007/978-3-030-50423-6_42)

11. Shelke A et al (2021) Chest X-ray classification using deep learning for automated COVID-19 screening. *SN Comput Sci* 2(4):300
12. Ahsan MM et al (2020) Covid-19 symptoms detection based on nasnetmobile with explainable ai using various imaging modalities. *Mach Learn Knowl Extract* 2(4):490–504
13. Chetoui M et al (2021) Explainable COVID-19 detection on chest X-rays using an end-to-end deep convolutional neural network architecture. *Big Data and Cognit Comput* 5(4):73
14. Kim S et al (2022) Deep learning in multi-class lung diseases' classification on chest X-ray images. *Diagnostics* 12(4):915
15. Sitaula C, Hossain MB (2021) Attention-based VGG-16 model for COVID-19 chest X-ray image classification. *Appl Intell* 51:2850–2863
16. <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database> . Accessed 15 Mar 2023
17. <https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset> . Accessed 15 Mar 23
18. <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia> . Accessed 15 Mar 23
19. Long W (2019) IEEE 23rd international conference on computer supported cooperative work in design (CSCWD). IEEE, pp 205–209. <https://doi.org/10.1109/CSCWD.2019.8791884>
20. He K et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
21. Demir, A, Yilmaz F, Kose O (2019) Early detection of skin cancer using deep learning architectures: resnet-101 and inception-v3. In: 2019 medical technologies congress(TIPTEKNO). IEEE, pp 1–4. <https://doi.org/10.1109/TIPTEKNO47231.2019.8972045>
22. Goutam K et al (2020) Layerout: freezing layers in deep neural networks. *SN Comput Sci* 1(5):295
23. Powers DMW (2020) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *ArXiv preprint arXiv:2010.16061*
24. Mohammadreza H, Doyle Thomas E, Reza S (2022) MLCM: multi-label confusion matrix. *IEEE Access* 10:19083–19095

# Stance Classification on FIFA World Cup Using Twitter Data



Aaquib Asrar, Susmita Das, and Sangita Dutta

**Abstract** The widespread influence of social media has led to an increase in online discussions on varied topics. International sports events with a worldwide following bring in a lot of online engagement between people with various viewpoints. Analyzing the stances in different public discourses from social media posts has gained research attraction. In this paper, we have tried to identify the stance of people on the FIFA World Cup 2022 from Twitter. Natural language processing techniques have been used for text preprocessing and compared different machine learning approaches to classify the Twitter text content into three stance classes of FAVOR, AGAINST, and NONE. We have observed that for our approach, support vector machine (SVM) provides the best result with good accuracy. Our approach is performing well on benchmark datasets.

**Keywords** Stance detection · Machine learning · Natural language processing

## 1 Introduction

In this period of the Internet, social media platforms are becoming dominant and replacing traditional online communication methods. An increasing number of people are obtaining news and information from social media. Sometimes, these platforms play a crucial role in determining public discourse regarding different issues and concepts. Researchers are gaining interest in understanding these online conversations and ascertaining the stance of the users on a certain topic from their posts on social media.

---

A. Asrar · S. Das · S. Dutta (✉)

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India  
e-mail: [duttasangita72@gmail.com](mailto:duttasangita72@gmail.com)

A. Asrar

e-mail: [2020csb058.aaquib@students.iests.ac.in](mailto:2020csb058.aaquib@students.iests.ac.in)

S. Das

e-mail: [susmitad900@gmail.com](mailto:susmitad900@gmail.com)

We have considered the recently concluded FIFA World Cup 2022 held in Qatar for our research. It has attracted spectators from all over the world and has one of the highest viewership and ratings. This global event became a primary subject of discussion on social media platforms. There are previous works on analyzing the bias [11] and sentiment of tweets [8] specific to Twitter. Detection of stances from social media regarding football has not been provided much attention.

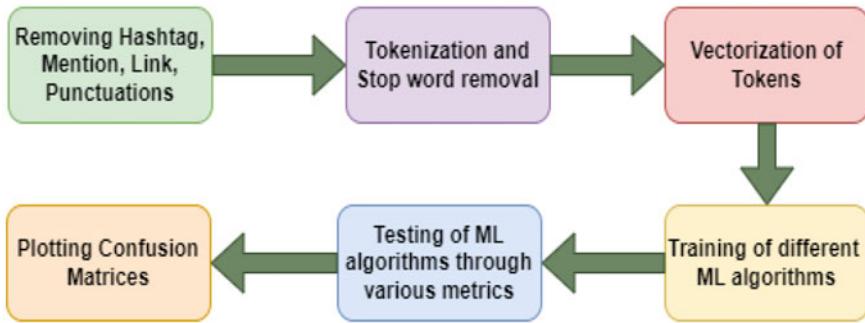
In our work, the stances of viewers toward the tournament have been analyzed as a whole using NLP and machine learning approaches. The rest of the paper contains related works in Sect. 2, proposed method in Sect. 3, result and analysis in Sect. 4, and conclusion in Sect. 5.

## 2 Related Works

Stance detection has been a trending area of research over the past few years. Dey et al. [5] have conducted the classification of tweets as neutral and non-neutral based on subjectivity. Hou et al. [10] studied the detection of stance with regard to COVID misinformation on Twitter. Grimminger et al. [6] examined political rivalry and general public stance detection using political tweets. Das et al. [4] analyzed the perception of people on climate change issues and the conference of parties(COP).

Haouari et al. [7] established the aim of identifying authorities' stance about rumors in tweets and made the first dataset available for the work focusing on rumors in Arabic. They investigated the efficacy of Arabic datasets already available for stance detection. Cao et al. [3] presented the overview of stance detection. Essential steps of the framework are described in depth as they first present a broad framework for stance detection. The three groups of cutting-edge stance detection techniques are feature-based techniques, deep learning techniques, and ensemble learning techniques. Also, the benefits and drawbacks of the current methodologies are examined. According to the survey results, hybrid neural network-based solutions are more effective than other methods. Price et al. [13] have researched the influence of Twitter in molding the relationship of different football clubs with their supporters. Hidayatullah et al. [9] conducted topic modeling to analyze the topics that are discussed in social media platforms regarding football news. Buongiovanni et al. [1] analyzed the creation of echo chambers on Twitter during Euro 2020 tournament in Italy regarding a specific topic. Burgers et al. [2] have also studied the bias in social media regarding football in live commentaries.

We have studied the interactions on Twitter during the whole FIFA World Cup 2022 Tournament and observed the stances of viewers.



**Fig. 1** Framework of proposed method

### 3 Proposed Method

We have proposed our novel method for stance classification using both natural language processing and machine learning methods.

#### 3.1 Data Collection and Labeling

We considered Twitter for our research, as most of the discussions between different individuals with a variety of opinions are exchanged on Twitter. Tweets were extracted using the Twitter API for the hashtag #FIFAWorldCup2022. A total of 15,372 tweets from 3rd December to 18th December 2022 have been extracted. Stances of all the tweets were labeled accordingly into three main categories: FAVOR, AGAINST, and NONE. To represent these stances in numbers, 1 for FAVOR, 0 or NONE, and -1 for AGAINST have been considered. In labeling their stances, the target was to understand the awareness and the viewpoints regarding FIFA World Cup 2022 on social media platforms. The tweet posts written in English have been considered, tweet posts in any other language were removed. The tweets have been vetted and considered the ones relevant to the tournament. The tweets with scoreline of any match have been removed as it would not make much sense in deciding the stance of the tweet. The framework of the proposed method is shown in Fig. 1.

#### 3.2 Preprocessing

The raw tweet data becomes difficult to handle, and the basic functions of natural language processing have been utilized for preprocessing. We tokenized the tweet text, removed all the stopwords and punctuations from the tweets. Emoticons have also been considered as they sometimes become significant in representing the stance of a particular tweet. The pronouns are converted to lowercase as it doesn't create much

difference in meaning, but as an example, “the” and “The” which are not pronouns signify different meanings. All the user mentions, hashtags, special characters, and URLs in the tweets were removed because they have no significance in deciding the stance of a tweet. Misspelt words were corrected and words that were in contracted form were normalized for ease of computation. The corresponding modules in NLP tools such as TweetNLP, Stanford CoreNLP, and NLTK were used for our preprocessing purposes.

### 3.3 Model Selection

In the proposed model, the preprocessed text data is taken as input, and the words or tokens are converted into vectors using the TF-IDF vectorizer.

**TF-IDF Vectorizer:** It is a vectorizer that combines the concepts of term frequency(TF) depicted in Eq. (1) and inverse document frequency(IDF) depicted in Eq. (2) to transform the text into vectors. TF is formulated as

$$\text{Term Frequency}_x (\text{TF}_x) = \frac{T_x}{F} \quad (1)$$

where  $T_x$  denotes the number of times the term  $x$  appears in the document and  $F$  denotes total number of terms in the document. IDF is formulated as

$$\text{Inverse Document Frequency}_x (\text{IDF}_x) = \log \left( \frac{N}{\text{DF}_x} \right) \quad (2)$$

where  $N$  denotes the total number of documents, and  $\text{DF}_x$  denotes the number of documents containing the term  $x$ .

TF-IDF score is calculated by Eq. (3):

$$\text{TF-IDF} = \text{TF} * \text{IDF} \quad (3)$$

**N-grams:** The n-grams of textual data are sequence of words or tokens occurring adjacent to each other. Uni-grams and bi-grams are considered in vectorizing the words. Consider an example tweet: **Argentina won the world cup**

This tweet contains uni-grams—“Argentina”, “won”, “the”, “world”, “cup” and bi-grams—(“Argentina”, “won”), (“won”, “the”), (“the”, “world”), (“world”, “cup”). Tri-grams, quad-grams, and bigger grams have not been considered as the previous tokens and their significance will get diminished.

After vectorization using Eq. (3), the dataset is divided into two sets, namely the training set and the testing set into an 80:20 ratio. Several machine learning algorithms are used on the training dataset in the initial phase. A logistic regression algorithm is applied to our preprocessed training dataset. A grid search has been done to get

the best setting for hyperparameters. The values of  $C$  are tuned and initialized class weight as balanced so that, no biasing occurs due to the unbalanced category sizes.

A random forest classifier is used on our training dataset and tuned some hyperparameters like  $n_{estimators}$  (*number of estimators*) to 500, criterion to “*entropy*”, and class weight to “*balanced*”.

Support vector machine(SVM) has been used to train our training set. The class weight parameter is also tuned and set it to “*balanced*”.

Bernoulli Naive Bayes(BernoulliNB) has also been used subsequently. The algorithm was applied on the training set and tuned the hyperparameter alpha( $\alpha$ ) and set its value to  $1e - 6$ . Moreover, decision tree classifier and support vector machine with RBF kernel were used and trained with the training data.

## 4 Results and Analysis

We evaluate our approach by implementing it on multiple datasets and used the metrics, precision, recall, and F1-score. Formulas for precision have been depicted in Eq. (4), recall is shown in Eq. (5), and F1-score is depicted in Eq. (6):

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (5)$$

$$\text{F1-score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (6)$$

where TP denotes true positive, FN denotes false negative, and FP denotes false positive.

### 4.1 Analysis on Benchmark Dataset

We have conducted evaluation of our approach on the benchmark dataset SemEval2016 [12] for stance detection. This benchmark dataset contains records based on different issues raised on Twitter. Tweets are categorized based on *Atheism*, *Climate Change*, *Feminist Movement*, *Abortion Legalization*, *Hillary Clinton* and *Donald Trump*. The precision, recall, accuracy, and F1-score for the three stances have been analyzed. As depicted in Table 1, our approach is providing great *precision* results for the stances in *FAVOR* and *AGAINST* for the topic of *climate change*. The *precision* results for the topic of *Atheism* are good, and for *Hillary Clinton*, *Feminist Movement* and *Abortion Legalization* results are quite satisfactory. The *recall* results

**Table 1** Precision, recall, F1-score, and accuracy on benchmark dataset

Topic	Stances	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Climate change	AGAINST	100	33	56	77
	NONE	68	79	75	
	FAVOR	79	71	80	
Atheism	AGAINST	71	93	78	74
	NONE	67	17	47	
	FAVOR	83	56	79	
Hillary Clinton	AGAINST	73	76	75	72
	NONE	67	65	62	
	FAVOR	68	48	72	
Feminist Movement	AGAINST	60	70	64	57
	NONE	18	8	11	
	FAVOR	45	48	47	
Abortion Legalization	AGAINST	66	76	71	64
	NONE	58	33	42	
	FAVOR	51	59	55	

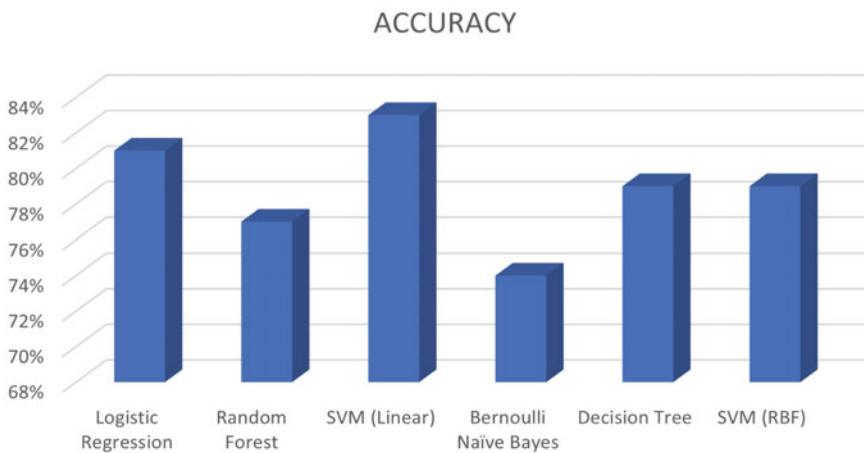
**Table 2** Comparison with the previous method

Approach	$F_{avg}$
Naive Bayes+ID3-(NER+TF-grams comb) [14]	68.6
SVM-uni-grams [12]	63.31
Our approach	69.33

obtained are also satisfactory for all three stances across all five topics. As depicted in Table 1, it is observed that the *F1-Score* is best for tweets having stance in *FAVOR*, while for tweets, having stance *AGAINST* is moderate in case of *climate change*. For *Atheism* and *Hillary Clinton*, tweets having stance *AGAINST* have comparatively better *F1-score*. The best *accuracy* of 77% is obtained for the topic of *climate change*, while values of *Hillary Clinton* and *Atheism* are also adequate. The lowest *accuracy* of 57% is observed in the case of *Feminist Movement* topic. Our work is compared with the previous method [14] with respect to the SemEval2016 dataset. As depicted in Table 2, it is observed that in comparison with the previous methods, our approach is providing good value for average F1-score( $F_{avg}$ ), and the results are satisfactory.

**Table 3** Accuracy and details of all classifiers

Classifiers	Hyperparameters	Accuracy (%)
Logistic regression	C = 10, class_weight = ‘balanced’	81
Random forest classifier	n_estimators = 500, criterion = ‘entropy’ , class_weight = ‘balanced’	77
Support vector machine (linear)	C = 1, class_weight = ‘balanced’	83
Bernoulli Naïve Bayes	alpha = 1e–6	74
Decision tree classifier	max_depth = 128, class_weight=‘balanced’	79
Support vector machine	Kernel = ‘rbf’, C = 10	79

**Fig. 2** Accuracy comparison of different classifiers

## 4.2 Analysis on Collected Dataset

We analyzed our approach for the FIFA World Cup 2022 tweets. After applying the logistic regression model, an accuracy of about 81% is obtained on the test set. In the case of random forest classifier, after training the model, the accuracy value lowers, and an accuracy of about 77% is obtained on the test set. In case of both decision tree classifier and SVM using RBF kernel, an accuracy of 79% is obtained on the test set which is little higher than random forest. In case of SVM, an accuracy of about 83% is estimated on our World Cup test set, which is the best accuracy obtained. For BernoulliNB, an accuracy of about 74% is evaluated on the test set which is comparatively lower than all the other models. The details about the classifiers and the hyperparameters used are summarized in Table 3. The accuracy of the classifiers has been depicted in Fig. 2.

**Table 4** F1-score for all classifiers

Classifiers	F1-score
Logistic regression	FAVOR -> 89%, NONE -> 72%, AGAINST -> 78%
Random forest classifier	FAVOR -> 87%, NONE -> 71%, AGAINST -> 68%
<b>Support vector machine (linear)</b>	<b>FAVOR -&gt; 90%, NONE -&gt; 73%, AGAINST -&gt; 81%</b>
Bernoulli Naïve Bayes	FAVOR -> 83%, NONE -> 66%, AGAINST -> 67%
Decision tree classifier	FAVOR -> 86%, NONE -> 69%, AGAINST -> 77%
Support vector machine	FAVOR -> 87%, NONE -> 70%, AGAINST -> 73%

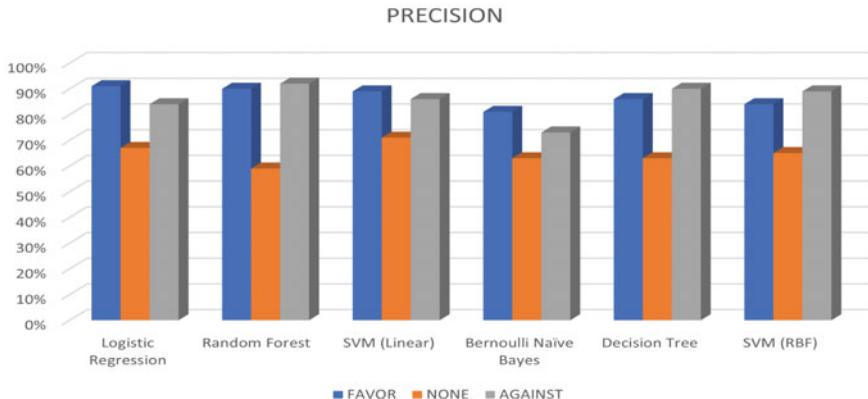
**Table 5** Precision and recall values for all classifiers

Classifiers	Precision	Recall
Logistic regression	<b>FAVOR -&gt; 91% NONE -&gt; 67% AGAINST -&gt; 84%</b>	FAVOR -> 88% NONE -> 78% AGAINST -> 72%
Random forest classifier	FAVOR -> 90% NONE -> 59% AGAINST -> 92%	FAVOR -> 84% <b>NONE -&gt; 87% AGAINST -&gt; 54%</b>
Support vector machine (linear)	FAVOR -> 89% <b>NONE -&gt; 71% AGAINST -&gt; 86%</b>	<b>FAVOR -&gt; 90% NONE -&gt; 76% AGAINST -&gt; 77%</b>
Bernoulli Naïve Bayes	FAVOR -> 81% NONE -> 63% AGAINST -> 73%	FAVOR -> 84% NONE -> 68% AGAINST -> 84%
Decision tree classifier	FAVOR -> 86% NONE -> 63% AGAINST -> 90%	FAVOR -> 85% NONE -> 78% AGAINST -> 68%
Support vector machine	FAVOR -> 84% NONE -> 65% AGAINST -> 89%	FAVOR -> 89% NONE -> 76% AGAINST -> 62%

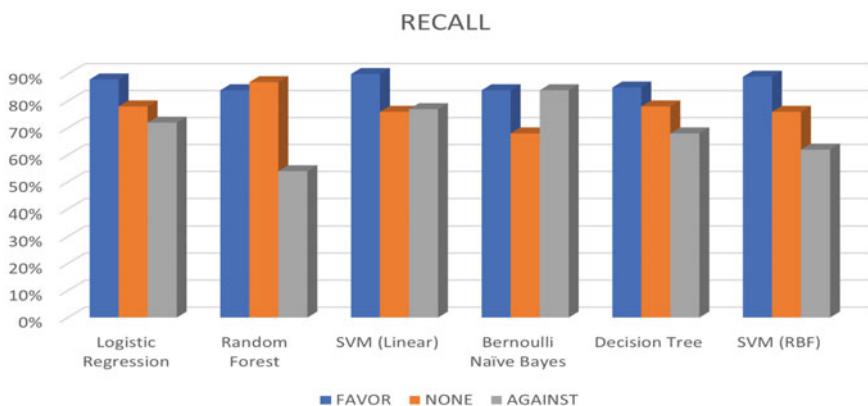
Considering the stance FAVOR, the best value of F1-score at 90% is obtained in the case of SVM(linear). Although logistic regression at 89% is quite close to SVM, the other classifiers linger behind. Bernoulli Naive Bayes performs comparatively worst with an F1-score of 83%.

For stance AGAINST, almost the same trend is observed with the best F1-score of 81% for SVM and the worst F1-score of 67% for Bernoulli Naive Bayes. The detailed F1-score for all the classifiers and respective three stances is depicted in Table 4.

When we evaluate our approach based on precision, logistic regression gives the best result of 91% for stance in FAVOR. But for recall, SVM again gives the highest value of 90%. But for stance AGAINST, the best precision value of 92% is given by random forest classifier and the highest recall value of 84% by Bernoulli Naive Bayes. The precision and recall values of different classifiers are portrayed in Table 5, and the highest values of both the metrics for all three stances are depicted in bold font. The comparison of the precision and recall values of different classifiers have been depicted in Figs. 3 and 4. The confusion matrices for the topics of climate change (Fig. 5a), Atheism (Fig. 5b), Hillary Clinton(Fig. 5c), Abortion Legalization



**Fig. 3** Precision comparison of different classifiers

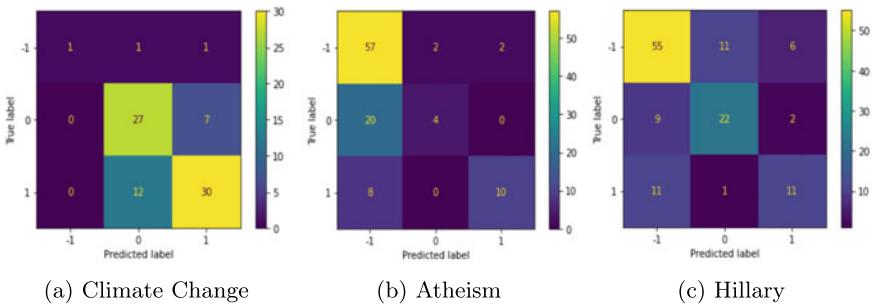


**Fig. 4** Comparison of recall values of different classifiers

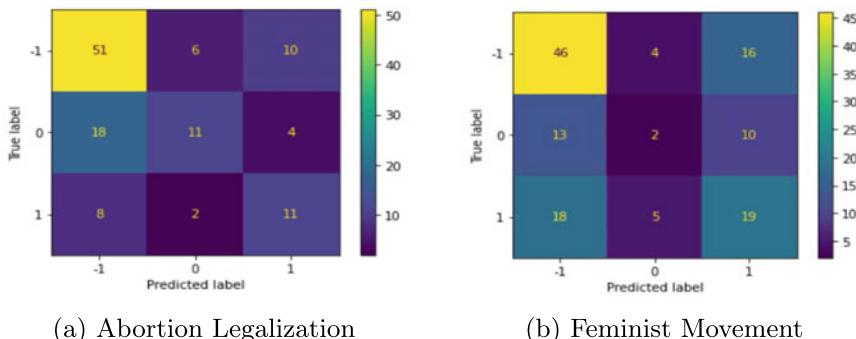
(Fig. 6a), and Feminist Movement (Fig. 6b) in the benchmark dataset are depicted in Figs. 5 and 6. These confusion matrices are obtained by implementing the SVM approach.

## 5 Conclusion

Stance classification is the major research topic discussed in this paper. We have collected our FIFA World Cup 2022 dataset from Twitter and preprocessed the raw data using natural language processing techniques for better handling of the information. Different machine learning classifiers like SVM, logistic regression, random forest , Bernoulli Naive Bayes, and decision tree classifies to detect the stance of the tweets



**Fig. 5** Confusion matrices using SVM approach



**Fig. 6** Confusion matrices on benchmark dataset using SVM

have been used. We compared the accuracy, F1-score, precision, and recall for all the classifiers and observed that SVM performs the finest among all the models. Our approach has been evaluated on benchmark dataset and obtained good results.

## References

1. Buongiovanni C, Candusso R, Cerretini G, Febbe D, Morini V, Rossetti G (2023) Will you take the knee? italian twitter echo chambers' genesis during euro 2020. In: Complex networks and their applications XI: proceedings of the eleventh international conference on complex networks and their applications: complex networks 2022-Volume 1. Springer, pp 29–40
2. Burgers C, Beukeboom CJ, Smith PA, van Biemen T (2023) How live twitter commentaries by professional sports clubs can reveal intergroup dynamics. Comput Human Behav 139:107528
3. Cao R, Luo X, Xi Y, Qiao Y (2022) Stance detection for online public opinion awareness: an overview. Int J Intell Syst. <https://doi.org/10.1002/int.23071>
4. Das S, Chakraborty S (2022) Perception of united nations climate change conference in social networks. In: 2022 IEEE 19th India council international conference (INDICON). IEEE, pp 1–6
5. Dey K, Shrivastava R, Kaushik S (2017) Twitter stance detection-a subjectivity and sentiment polarity inspired two-phase approach. In: 2017 IEEE international conference on data mining workshops (ICDMW). IEEE, pp 365–372

6. Grimminger L, Klinger R (2021) Hate towards the political opponent: a twitter corpus study of the 2020 us elections on the basis of offensive speech and stance detection. ArXiv preprint [arXiv:2103.01664](https://arxiv.org/abs/2103.01664)
7. Haouari F, Elsayed T (2023) Detecting stance of authorities towards rumors in Arabic tweets: a preliminary study. ArXiv preprint [arXiv:2301.05863](https://arxiv.org/abs/2301.05863)
8. Hegde SU, Zaiba A, Nagaraju Y et al (2021) Hybrid cnn-lstm model with glove word vector for sentiment analysis on football specific tweets. In: 2021 international conference on advances in electrical, computing, communication and sustainable technologies (ICAECT). IEEE, pp 1–8
9. Hidayatullah AF, Pembrani EC, Kurniawan W, Akbar G, Pranata R (2018) Twitter topic modeling on football news. In: 2018 3rd international conference on computer and communication systems (ICCCS). IEEE, pp 467–471
10. Hou Y, van der Putten P, Verberne S (2022) The covmis-stance dataset: stance detection on twitter for covid-19 misinformation. ArXiv preprint [arXiv:2204.02000](https://arxiv.org/abs/2204.02000)
11. Kim Y, Billings AC (2017) A hostile sports media? perceived nationalism bias in online sports coverage. Electron News 11(4):195–210
12. Mohammad S, Kiritchenko S, Sobhani P, Zhu X, Cherry C (2016) SemEval-2016 task 6: detecting stance in tweets. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp 31–41
13. Price J, Farrington N, Hall L (2013) Changing the game? the impact of twitter on relationships between football clubs, supporters and the sports media. Soccer and Soc 14(4):446–461
14. Tun YM, Myint PH (2019) A two-phase approach for stance classification in twitter using name entity recognition and term frequency feature. In: 2019 IEEE/ACIS 18th international conference on computer and information science (ICIS). IEEE, pp 77–81

# Machine Learning-Based Phishing Detection in Heterogeneous Information Network



Priti Halder, Ankan Mallick, Shantonu Debnath, and Malay Kule

**Abstract** Phishing is a kind of social engineering technique that lures a person into sharing sensitive details such as social security numbers, pass codes and pins. There are a variety of phishing detection techniques in the emerging world of technology. List-based techniques or the site content-based techniques could not cope up with the uninterrupted development of phishing methodology. Hence, in this paper, an enhanced technique is proposed which constructs a heterogeneous information network (HIN) using the hyperlinks from a webpage and extracts features from the HIN which in turn is used by a classifier to classify the webpage as phishing or legal. Experiments are carried out using a dataset published by Phishpedia to validate the proposed method. The performance of the proposed method was compared among different classification algorithms in terms of different quality metrics, namely recall, F1-score, precision and accuracy.

**Keywords** Heterogeneous information network · Phishing · Machine learning · Security · Authenticity

## 1 Introduction

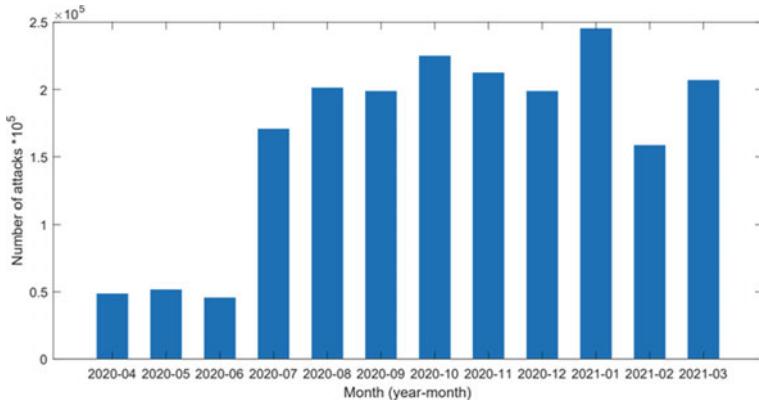
While using the Internet, people come across many different types of threats. Phishing is one of the most influential ones. Phishing is very much the same as catching fish in a river. The only difference here is the fact that here the hackers are trying to get hold of the information of the victims [1]. It is a kind of social engineering attack that collects information from end-users using fake websites that look real. A report was launched by the anti-phishing working group (APWG) [2] which describes the phishing attacks reached its peak in January 2021 as shown in Fig. 1.

---

P. Halder (✉) · A. Mallick · S. Debnath · M. Kule

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur 711103, India

e-mail: [prityhalder98@gmail.com](mailto:prityhalder98@gmail.com)



**Fig. 1** APWG report

Research on various anti-phishing techniques have been done in the recent era. These techniques can be categorised as list-based techniques, visual-similarity-based, side channel-based and machine learning techniques. List-based techniques generally work with the white-list technique and the blacklist techniques. White-list [3]-based method detects a phishing website by matching parallelism between the target webpage and the items in the list. If the similarity index is below a particular level, it is declared a phishing website. As for the blacklist [4, 5] method if the similarity index is higher than the certain level, it is declared as malicious. The main drawback of the list-based techniques is that it cannot prevent zero-day attacks. Visual-similarity-based methodology believes that the webpages of two different companies cannot be the same. If the visual appearance of two distinct sites is the same, but their URLs are dissimilar, then one website is declared malicious [6]. These methodologies can be avoided by changing the positions or components of the elements in the webpage. Some HIN-based algorithms like Hin Phish [7] use modified authority ranking algorithm [8] to obtain a quantified score to measure the authenticity of the target webpage. Although these approaches show a very good accuracy, they are infeasible for real-time and client-side implementation. Along with these, there are also some drawbacks like computational burden, third-party dependencies, etc. The main reason for failure of traditional approaches are distinct characteristics and single focus which leads to reduced focus to the associations among malign websites.

In this paper, a phishing detection methodology is proposed, which extracts hyper-links from the html source code of the target webpage and constructs a heterogeneous information network (HIN). An information network can have one or more than one type of objects and one or more than one kind of links. It is known as heterogeneous network (HIN) when there is more than one object type or more than one link type. Else, it is a homogeneous information network. Thirty-one features are extracted from the HIN, and a feature vector is prepared. Extensive experiments are conducted using different machine learning methodologies to compute the effectiveness of the

suggested features. The outcome exhibit that the proposed process clearly recognises the phishing websites as it has an elevated true positive rate and very less false negative rate.

Remaining paper is divided as follows: Sect. 2 describes the different works done on this domain, Sect. 3 describes our suggested approach and Sect. 4 describes the experimental results followed by conclusion in Sect. 5.

## 2 Literature Survey

Shahrivari et al. [9] discussed a method which is among the most successful technologies for finding these malignant activities, i.e. machine learning. Many of the hacking have some ordinary features that can be detected by machine learning techniques. The article analyses the findings of multiple machine learning techniques for predicting malign websites.

Mughaid et al. [1] proposed a detection architecture using ML methods by dividing the dataset to upskill the detection model and approve the findings utilising the trial data, to apprehend built-in features of the email text and other characteristics to be categorised into malignant and benign using three types of datasets.

Cao et al. [3] discussed a novel method called Automated Individual White-List (AIWL). AIWL naturally attempts to keep a white-list of all known Login User Interfaces (LUIs) of webpages. When an end-user tries to enter their secret data to an LUI which is absent in the white-list, an alert will be sent to the user for possible attack.

Prakash et al. [4] discussed PhishNet that exploits the fact that attackers often employ simple modification in the URL using two parts. In the first part, five heuristics are explained to finalise a simple combination of known malign sites to find out new malign URLs. The next part contains an approximate matching algorithm.

Rao et al. [5] discussed about an escalated blacklist method which uses key distinguishing characteristics taken out from the binary code of the webpage for the detection of malign webpages.

Jain et al. [6] reported an analysis of different kinds of phishing attacks, modern visual-similarity-based methods for detecting phishing and their analytics.

In [7], a malign website detecting procedure known as the HinPhish has been proposed. In the algorithm, various components are formed from the different types of links and a heterogeneous network is formed. The algorithm then assigns some weight to different categories of the links and uses them to calculate the phish score which apparently tells the quality of the webpage.

Shi et al. [10] presented the issue of creating clusters for a specific kind of objects, along with ranking data for all kinds of objects on the basis of these clusters in a HIN and propose the clustering structure called RankClu.

### 3 Proposed Method

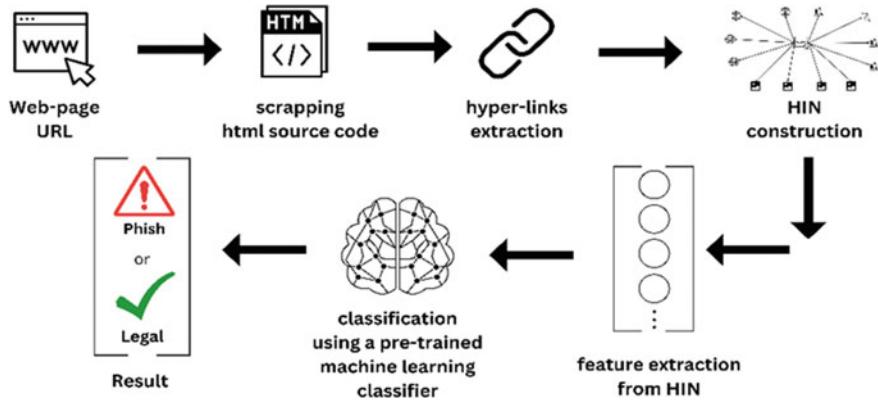
Static webpages on the Internet are almost obsolete today. Most of the webpages are now interactive. But unlike static webpages, these interactive webpages cannot exist independently. They need to communicate with other hosts to share information and load resources. For this purpose, webpages are now connected with different other objects on the Internet like other webpages, images, CSS files, external scripts, etc., through hyperlinks. These objects have different roles to perform in respect to the concerned webpage. The hyperlinks connecting these objects are also of different types. These connected objects and links of a webpage form an information network on the Internet. Considering the objects as nodes and the links among them as edges, the network of a webpage can be modelled as a heterogeneous information network.

To completely mimic a webpage, a phisher not only needs to mimic that particular webpage but also needs to replicate its entire information network on the Internet to keep the relationships intact. This significantly increases the cost of attack. To make the attack cost effective, the phishers in most of the cases mimic only the format and functionality of the concerned webpage and fit it within the existing information network of the actual webpage. This leads to changes in relationship among the objects and creates stark differences between the HIN of a phishing and a legal webpage, e.g., let us say there is a webpage P which loads some images on it. These images are hosted on the same host as the webpage P. The webpage P has a local relationship with the image objects as they are hosted at the same place. Now, a phisher creates a duplicate webpage P' duplicating the webpage P and hosts it on another host. But the images on P' are loaded from the previous host where the actual page is located. Now for P', the relationship of it with the images becomes foreign as they are hosted on a different host which means although P and P' look and work in the same way, their information network on the Internet is completely different. The proposed method analyses the HIN of a webpage and classifies it as phishing or legal using pre-trained classifiers.

In this paper, the structure of the methodology is divided into three parts. In the initial part, the HIN of a webpage is constructed using the hyperlinks embedded in its html source code. In the next phase, essential features are taken out from the HIN and a feature vector is prepared. In the third phase, the feature vector is given as input to a pre-trained ML classifier, which classifies an input as phish or legal. Figure 2 shows the workflow of the proposed algorithm.

#### 3.1 HIN Construction

Webpages use hyperlinks to connect to other objects. These hyperlinks can be found in four different.



**Fig. 2** Workflow diagram of the proposed approach

Tags:

- <a> tag: connection to a new page.
- <img> tag: contains location of the image file.
- <link> tag: defines relation between current document and an external resource.
- <script> tag: defines the location of the external script.

### 3.1.1 Object Classification

The objects indicated by those extracted links are classified into three categories.

- Page object: connected by links extracted from <a> tag.
- Resource object: connected by links extracted from <link>, <img> tag.
- Script object: connected by links extracted from <script> tag.

Note that the page objects can connect to further resource objects, but resource and script objects are terminal.

### 3.1.2 Link Classification

The extracted links are classified into four types.

- Null link: the link which leads to an empty page.
- Relative link: hostname of link is void.
- Local link: hostname of link and the concerned webpage is the same.
- Foreign link: hostname of link and concerned webpage is different.

### 3.1.3 Network Construction

Taking the objects as nodes and their links with the target webpage as edges, a directed graph is constructed. As the page objects can further connect to other objects, in this way, theoretically, the network can be extended to infinity adding objects at each subsequent level. A larger network will be able to capture more nuanced features of the HIN. In reality though, handling such a big amount of data is tough.

To extend the HIN beyond the neighbours of the target, the hyperlinks of each of the neighbouring webpages are needed. So, it is needed to crawl each webpage separately and each crawling involves a network delay; the runtime of the algorithm increases beyond feasible limits. As one of the goals of our approach is to detect phishing in real time and on the client side, extending the network beyond the depth of zero will be infeasible. So, in our work, the network is restricted within the neighbours of the target webpage only.

Instead of taking each object separately, those having links of the same domain name are clubbed into one object. For an object category, as all local links have the same domain name as the target page, all of them are replaced by one object. In case of relative links, the domain name is missing, so they are also clubbed together. Multiple objects can be possible in case of foreign links as there can be multiple domain names. Null links are considered one object.

In this way, each category of object has at least four objects:

- I. one local object.
- II. one relative object.
- III. one null object.
- IV. one or more than one foreign object.

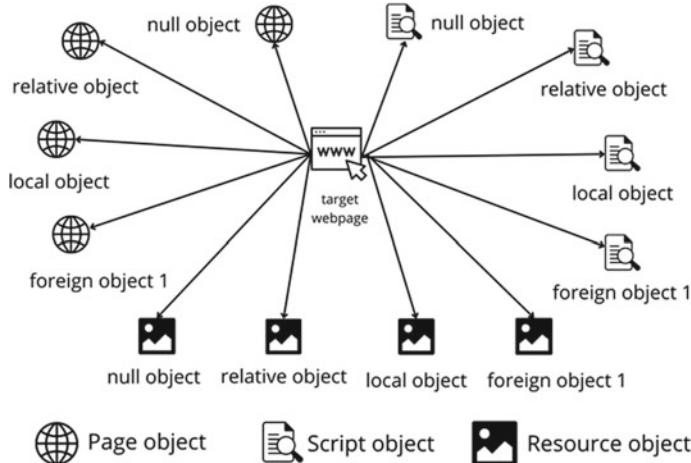
Figure 3 shows a diagram of HIN of a webpage.

## 3.2 Feature Selection

This section describes the features used to help identify the malign website.

### 3.2.1 Trust Score

Phish and legal websites develop very different network patterns. The legal websites generally have almost equal proportions of all four types of links. Phishing websites tend to have a very low proportion of local and relative links. Moreover, as the local links of the original website become foreign links in the phishing website, the majority of foreign links generally have a common domain name (of the original website) whereas, in legal websites, one foreign domain name generally has very a few links. Based on these features and the pattern of the network, a score is calculated for the target domain, namely the trust score. The score ranges from  $-1$  to  $1$ . A website



**Fig. 3** Diagram of HIN

having a lesser trust score is more likely to be a phishing website. Each object in the HIN is assigned a value same as the number of links clubbed into that object, e.g., the value of a foreign page object of domain name ‘d’ will be the number of foreign links connecting a page object of domain name ‘d’ with the target. Various kinds of objects and links contribute differently to the trust score. Higher foreign and null links are likely to decrease the score whereas local and relative links are likely to increase. Considering this assumption, the foreign and null links in the network are assigned a negative weight. On the other hand, local and relative links are assigned positively.

The weight of a link signifies the effect it has on the trust score. As it is assumed that more foreign and null links are traits of a phishing website, these types of links should have a negative effect on the trust score. On the other hand, relative and local links should have a positive effect. The weight should be set in such a way that an average page (a page having number of links equal to the average of the dataset) has a trust score zero. In that case, pages with a legal bias will have a positive trust score and pages with phishing bias will have a negative one. The average numbers of the four types of links for various kinds of objects are given in Table 1.

The average number of links is different for different types of links. The magnitude of weight should scale the numbers to equal. So, reciprocal of the average number

**Table 1** Average number of links

Object type	Null link	Relative link	Local link	Foreign link
Page	2.18	2.74	0.56	3.80
Resource	5.25	12.86	1.35	21.92
Script	2.94	17.44	2.44	5.61

**Table 2** Weight of different links

Object type	Link type	Weight
Page object	Null	-0.46
	Relative	0.36
	Local	1.79
	Foreign	-0.26
Resource object	Null	-0.20
	Relative	0.08
	Local	0.74
	Foreign	-0.05
Script object	Null	-0.34
	Relative	0.06
	Local	0.41
	Foreign	-0.18

of each type of link of each object is taken as its weight. Weight vectors PW, RW, SW, W were created according to Table 2.

$PW_i$ : weight of link between  $i$ th page object and target webpage.

$RW_i$ : weight of link between  $i$ th resource object and target website.

$SW_i$ : weight of link between  $i$ th script object and target website.

$W_i$ : weight of link between  $i$ th object (all types of objects) and target website.

Value vectors PV, RV, SV and V were also created:

$PV_i$ : Value of the  $i$ th page object.

$RV_i$ : Value of the  $i$ th resource object.

$SV_i$ : Value of the  $i$ th script object.

$V_i$ : Value of the  $i$ th object (including all object).

Trust score for the entire network is calculated as given in Eq. (1):

$$F0 = \frac{\sqrt[3]{\sum \{(V_i \times W_i)^3\}}}{\sum V_i}, \quad \sum V_i \neq 0 \\ = 0, \quad \text{otherwise} \quad (1)$$

Three different scores for the three kinds of objects are also to be calculated as given in Eqs. (2)–(4).

$$F1 = \frac{\sqrt[3]{\sum \{(PV_i \times PW_i)^3\}}}{\sum PV_i}, \quad \sum PV_i \neq 0 \\ = 0, \quad \text{otherwise} \quad (2)$$

$$F2 = \frac{\sqrt[3]{[\sum \{(RV_i \times RW_i)^3\}]}}{\sum RV_i}, \quad \sum RV_i \neq 0 \\ = 0, \quad \text{otherwise} \quad (3)$$

$$F3 = \frac{\sqrt[3]{[\sum \{(SV_i \times SW_i)^3\}]}}{\sum SV_i}, \quad \sum SV_i \neq 0 \\ = 0, \quad \text{otherwise} \quad (4)$$

where F0 is the trust score for all objects in the network and F1(2), F2(3), F3(4) are trust scores for only page, resource and script objects, respectively.

### 3.2.2 Concentration of Foreign Link

The foreign links in the network of a phishing website can be of a single domain name or multiple domain names. As the local links of the original website become foreign in phishing websites, most of its foreign links tend to have a common domain name. This feature is missing in legal websites. Their foreign links of different external websites have different domain names. So more the foreign links tend to concentrate on a single domain name, the more likely the website is phishing. To leverage this characteristic, a new feature namely ‘concentration ratio of foreign links’ is proposed.

Following vectors are created:

- PVF<sub>i</sub>: Value of the *i*th foreign page object.
- RVF<sub>i</sub>: Value of the *i*th foreign resource object.
- SVF<sub>i</sub>: Value of the *i*th foreign script object.
- VF<sub>i</sub>: Value of the *i*th foreign object (including all objects).

They are calculated as given in Eqs. (5)–(8):

$$F4 = \frac{\sqrt[3]{[\sum \{(PVF_i)^2\}]}}{\sum PVF_i}, \quad \sum PVF_i \neq 0 \\ = 0, \quad \text{otherwise} \quad (5)$$

$$F5 = \frac{\sqrt[3]{[\sum \{(RVF_i)^2\}]}}{\sum RVF_i}, \quad \sum RVF_i \neq 0 \\ = 0, \quad \text{otherwise} \quad (6)$$

$$F6 = \frac{\sqrt[3]{[\sum \{(SVF_i)^2\}]}}{\sum SVF_i}, \quad \sum SVF_i \neq 0$$

$$= 0, \quad \text{otherwise} \quad (7)$$

$$F7 = \frac{\sqrt{[\sum \{(VF_i)^2\}]}}{\sum VF_i}, \quad \sum VF_i \neq 0 \\ = 0, \quad \text{otherwise} \quad (8)$$

where F4(5), F5(6) and F6(7) are concentration ratio of foreign links for page, resource and script objects, respectively, and F7(8) is the concentration ratio of foreign links for all objects.

### 3.2.3 Number of Links

Phishing websites have more null and foreign links as compared to local and relative links. So, the numbers of the four types of links for each object are used as features.

F8, F9, F10 and F11 are the number of null, relative, local and foreign links connecting page objects, respectively.

F12, F13, F14 and F15 are the number of null, relative, local and foreign links connecting resource objects, respectively.

F16, F17, F18 and F19 are the number of null, relative, local and foreign links connecting script objects, respectively.

### 3.2.4 Proportion of Links

The composition of different types of links in the link pool is different for phishing and legal websites. Moreover, phishing websites tend to have a lesser total number of links compared to legal websites. Using only a number of links as features may mislead the algorithm. So, the proportions of the four types of links for each object are used as features.

F20, F21, F22 and F23 are the proportion of null, relative, local and foreign links connecting page objects, respectively.

F24, F25, F26 and F27 are the proportion of null, relative, local and foreign links connecting resource objects, respectively.

F28, F29, F30 and F31 are the proportion of null, relative, local and foreign links connecting script objects, respectively.

## 3.3 Algorithms

The algorithms to calculate the trust score and concentration ratio of foreign links are elaborated below.

### 3.3.1 Trust Score Computation

Input: Value\_vector: VT; Weight\_vector: WT Output: Trust\_score: FT  
// Initialization

1. ST = 0
2. Total = 0
3. for i in len(VT) do
4. ST += pow (VT[i] \* WT[i], 3)
5. Total += VT[i]
6. FT = pow (ST, 1/3) / Total
7. return FT

### 3.3.2 Concentration Ratio of Foreign Links Calculation

Input: Value\_vector: VFT Output: Concentration\_ratio: FT  
// Initialization

1. Cr = 0
2. Total = 0
3. for i in len (VFT) do
4. Cr += pow (VFT[i] \*,2)
5. Total += VFT[i]
6. FT = pow (Cr,1/2)/Total
7. return FT

## 3.4 Limitations

This approach requires the html source code of the concerned website. Though a sufficient number of phishing links is available from different sources, they are mostly inactive due to obvious reasons and their html source codes are not available due to limitation of computation power, and sufficient number of legal pages could not be scrapped. It is possible to find more accurate weights for the links of the network using further statistical analysis and machine learning techniques.

## 4 Experimental Result

The dataset, evaluation metrics and the results of the experiment are elaborately explained.

**Table 3** Dataset description

	Phish sample	Benign sample
Total samples	31,669	30,068
Invalid samples	4381	11,152
Valid samples	27,228	18,916

## 4.1 Dataset

The algorithm is evaluated on a dataset of phish and legal websites. The html source codes of phishing websites are collected from the recent dataset published by Phish-pedia. The html source codes of legal websites are scrapped from the URLs provided by top-rank Alexa list. Some links were unresponsive, so we could not get their html code. Some html source codes could not be processed. The details of the final dataset used are given in Table 3.

## 4.2 Evaluation Metrics

The metrics given in Eqs. (9)–(12) are used to measure the performance of the algorithm:

True positive (TP) = Number of websites correctly classified as phish in test dataset.

False positive (FP) = Number of websites incorrectly classified as phish in test dataset.

True negative (TN) = Number of websites correctly classified as legal in test dataset.

False negative (FN) = Number of websites incorrectly classified as legal in test dataset.

$$\text{Precision}(P) = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (9)$$

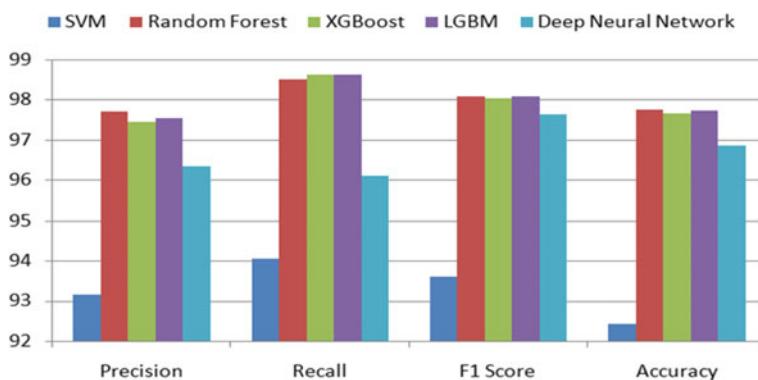
$$\text{Recall}(R) = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (10)$$

$$\text{F1 Score} = \frac{2\text{PR}}{(\text{P} + \text{R})} \quad (11)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} \quad (12)$$

**Table 4** Test results for different classification algorithms

Classifier	Precision	Recall	F1 score	Accuracy
SVM	93.15	94.05	93.60	92.42
Random Forest	<b>97.70</b>	98.51	<b>98.10</b>	<b>97.75</b>
XGBoost	97.46	98.62	98.04	97.67
LGBM	97.54	<b>98.64</b>	98.09	97.73
Deep Neural Network	96.35	96.10	97.63	96.86

**Fig. 4** Comparison of results

### 4.3 Results

A classifier is trained on the dataset using different classification algorithms. For training, k-fold cross-validation technique is used. In this technique, the used dataset is divided into k groups. The classifier is trained on (k-1) groups and tested on the remaining k-th group. This process is repeated, taking each of the k groups as the test group. All of the features in the dataset are normalised before training. The evaluation metrics are calculated from the test predictions. Results are given in Table 4. The best value of each metric is marked in bold.

Figure 4 shows the comparison between the results of the three classifiers, namely SVM, random forest, XGB, LGBM and Deep Neural Network. As per the results, the classification is better using the Random Forest classifier.

## 5 Conclusion

List-based techniques are traditional detection methods, but these methods fail to detect zero-day attacks. Machine learning methods are being used to escalate the efficiency of detection and decrease the misclassifications. In this work, a machine

learning-based method is proposed which classifies phishing websites by extracting features from its hyperlinks and the heterogeneous information network it forms. The proposed method is a fully client-side adaptable, can work in real-time and does not depend on any third-party services. Additionally, the semantic data of relationships and objects depending on HIN increases the cost of attack for attackers. The accuracy can be further improved by using a larger dataset. In due course, we aim to upgrade the formation of HIN and confirm the feasibility of the proposed method on a real-world larger data.

## References

1. Mughaid A, AlZu'bi S, Hnaif A, Taamneh S, Alnajjar A, Abu Elsoud EA (2022) An intelligent cyber security phishing detection system using deep learning techniques. Cluster Comput. <https://doi.org/10.1007/s10586-022-03604-4>
2. APWG. Phishing activity trends report. Available online: [https://docs.apwg.org/reports/apwg\\_trends\\_report\\_q1\\_2021.pdf](https://docs.apwg.org/reports/apwg_trends_report_q1_2021.pdf). Accessed 11 Aug 2021
3. Cao Y, Han W, Le Y (2008) Anti-phishing based on automated individual white-list. In: Proceedings of the 4th ACM workshop on digital identity management, Alexandria, WV, USA, 31 October 2008; Association for Computing Machinery, New York, NY, USA, pp 51–60
4. Prakash P, Kumar M, Kompella RR, Gupta M (2010) Phishnet: predictive blacklisting to detect phishing attacks. In: Proceedings of the 2010 Proceedings IEEE INFOCOM, San Diego, CA, USA, 14–19 March 2010, pp 1–5
5. Rao RS, Pais AR (2017) An enhanced blacklist method to detect phishing websites. In: Lecture notes in computer science, Proceedings of the International conference on information systems security, Mumbai, India, 16–20 December 2017. Springer, Cham, Switzerland, pp 323–333
6. Jain AK, Gupta BB (2017) Phishing detection: analysis of visual similarity based approaches. Hindawi Secur Commun Netw 2017:20. Article ID: 5421046. <https://doi.org/10.1155/2017/5421046>
7. Guo B, Zhang Y, Xu C, Shi F, Li Y, Zhang M (2021) HinPhish: an effective phishing detection approach based on heterogeneous information networks. Appl Sci 11:9733. <https://doi.org/10.3390/app11209733>
8. Sun Y, Han J, Zhao P, Yin Z, Cheng H, Wu T (2009) Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In: Proceedings of the 12th International conference on extending database technology: advances in database technology, Saint Petersburg, Russia, 24–26 March 2009; pp 565–576
9. Shahrivari V, Darabi MM, Izadi M (2020) Phishing detection using machine learning techniques. <https://arxiv.org/abs/2009.11116>
10. Shi C, Li Y, Zhang J, Sun Y, Yu PS (2017) A survey of heterogeneous information network analysis. IEEE Trans Knowl Data Eng 29(1)

# Unified Feature Extraction for Handwritten and Natural Image Characters: A Case Study Using Bank Legal Check Amount and Vehicle License Plate Number Recognition



M. Arun, S. Arivazhagan, P. Sivaramapandian, and S. Sivavakisan

**Abstract** Even though the recognition of printed characters is considered to be a solved problem in the field of pattern recognition, similar methodologies produce unexpected results when applied to natural image characters and handwritten texts. This is because natural images have more characteristics than printed texts do, such as different angles and font styles, as well as background and foreground colors that affect the borders of shapes in natural images. Additionally, handwritten texts do not always have the same size, width, and orientation due to differences in writing styles, writing instruments, and other factors. In this research, a “unified” character recognition model based on traditional machine learning as well as deep learning method has been used to address the problems of bank check legal amount and automatic license plate recognition. Using these unified models, we have achieved accuracy above 85% for both the applications.

**Keywords** Bank check · License plate · Histogram of gradients · Chars74K · CNN · CVL single digit · DIGI-Net · Zoning

## 1 Introduction

The basic building elements of text are characters and they are essentially symbols that represent information (alphabets, digits, and special characters). The process by which a computer recognizes and converts the letters, digits, and symbols in a document or image into a digital format that can be read by any machine is termed as

---

M. Arun (✉) · S. Arivazhagan · P. Sivaramapandian · S. Sivavakisan  
Department of Electronics and Communication Engineering, Mepco Schlenk Engineering  
College, Sivakasi, Tamil Nadu 626005, India  
e-mail: [arun@mepcoeng.ac.in](mailto:arun@mepcoeng.ac.in)

S. Arivazhagan  
e-mail: [sarivu@mepcoeng.ac.in](mailto:sarivu@mepcoeng.ac.in)

character recognition. Recognition of “machine-printed text” is regarded as a problem that has been addressed by modern systems. Recognizing handwritten characters is difficult due to the enormous variation in its shape, size, pace of writing, character thickness resulting from the writing styles of different persons and persons writing at different instances. Text recognition in the natural images is also more difficult than printed texts due to the high intra-class variability due to image acquiring conditions (lighting, resolution, blurriness, etc.), a variety of fonts, multiple deformations, and a complicated background.

Development of character recognition systems designed for handwritten characters will significantly advance the automation processes of sorting postal mails [1], processing handwritten applications, recognizing bank checks, processing census data, digitizing the palm leaf and historical documents, etc. Similarly, the widespread usage of smartphones and surveillance cameras has created an enormous demand for natural scene character recognition. This natural scene character recognition is extremely useful for the understanding of text in images for computer vision applications such as image comprehension, technological aids for the blind, automatic number plate recognition, sign recognition, navigation, and geocoding.

Lei et al. [2] classified handwritten characters employing directional string features and a nearest neighbor matching classifier. Aradhya et al. [3] implemented handwritten digit recognition using the Radon Transform in the MNIST database and Kannada handwritten numerals. Ali and Foroosh [4] designed a character recognition system for natural environments using Histogram of Oriented Gradients (HOG) and Shape Contexts (SCs). Campos et al. [5] employed the bag-of-visual-words feature with SVM Classifier to construct a system for automatic text interpretation in natural images.

Using a global feature scheme and an HMM module, Guillevic and Suen [6] created a system for legal amount recognition on bank checks. Using hybrid architecture, Palacios et al. [7] built a system for bank check recognition. Using HOG and GLCM features, Ghosh et al. [8] have recognized the handwritten text on bank checks. Ozbay and Ercelebi [9] have created an Automatic Vehicle Identification system that uses statistical template matching for license plate character recognition. Kim et al. [10] have devised an algorithm that tackles the issues that arise during the recognition of a vehicle license plate through closed-circuit televisions (CCTVs).

We have created a unified model for handwritten digit recognition on bank checks and natural imagery-based car license plate number detection as a single framework. We employed the IDRBT dataset specifically for check amount detection, and to the authors’ knowledge, no other research effort has recognized check amounts, as this dataset was primarily developed for classifying the ink kinds used in checks. We extracted the digitized amount from the bank check and the license plate number from the vehicle’s license plate. The retrieved images are then segmented and classified using both classic SVM-based techniques and deep learning techniques. The proposed unified spatial feature extraction is a very simple framework with less complexity, and the proposed DIGI-Net CNN is also very memory efficient with lesser number of parameters.

Section 2 describes the materials and methods used in this research work. Section 3 deals with the experiment analysis and discussions. Finally, Sect. 4 concludes the research work.

## 2 Materials and Methods

This research work uses two datasets for the application part and two datasets for the model training part. For the check amount identification, we have used the Axis Bank checks from IDRBT dataset. For car license plate number recognition, we have used the Mercosur license plate dataset. The images from “CVL Single digit” dataset are used for the training of check amount recognition, and 36 classes (A–Z and 0–9) of natural image characters of Chars74K dataset are used for the training of license plate number recognition.

### 2.1 IDRBT Dataset

IDRBT [11] has compiled a database of photos of bank checks with diverse textures and ink colors. As a source document, 112 check leaves from four distinct Indian banks are utilized. Seven different black and blue pens are used to replicate the differences in pen ink on check leaves. To avoid writing-related bias, nine separate volunteers are actively contributed to the production of the dataset by writing the checks. Fourteen different pens are used by nine different persons to generate the ink data, for a total of 126 pen volunteer combinations. In practice, extra words are added to the source material using pens of the same color. This research utilizes the IDRBT database for the check amount recognition problem. This dataset contains 87 Axis Bank checks that are utilized in the recognition challenge. This dataset was previously used to train a set capable of detecting the pen and ink used to write a bank check’s name and amount. This dataset has never been utilized for project-related work such as handwritten digit recognition; rather, it is employed for recognition in this study.

### 2.2 Mercosur License Plate Dataset

The Mercosur license plate generator [12] was designed to produce fictitious license plate images that are compliant with the new standard and have sufficient variation for practicing automatic license plate recognition. This category contains images that have been altered in some way as a result of physical barriers or environmental factors. For the purpose of recognition, we utilized 166 photographs of license plates taken from the dataset. This ensures that the photos of the license plate numbers



**Fig. 1** Sample images of **a** IDRBT check dataset and **b** Mercosur license plate dataset

contain both real camera acquired photographs as well as images that have been created synthetically.

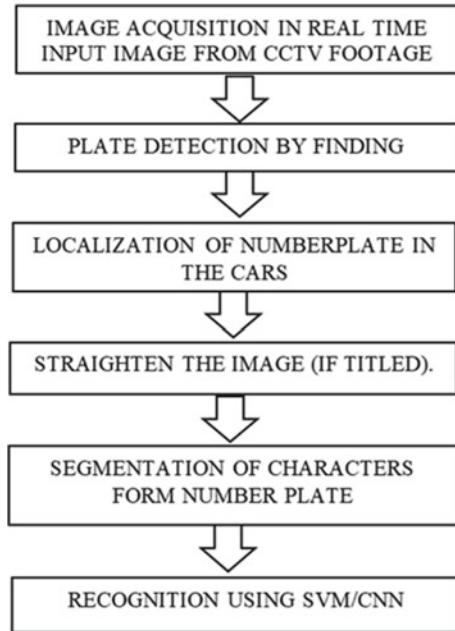
Figure 1 displays representative images from the check and license plate databases.

## 2.3 Methodology

The steps involved in extracting the check amount and license plate number and recognition from the dataset images are given as a flow diagram in Fig. 2 and the steps are explained below.

### 2.3.1 Preprocessing

Converting the RGB image into a grayscale image is the first step in preprocessing. The primary goal of color conversion is to limit the number of colors available thereby reducing the color complexity of the image. The next step in preprocessing involves image filtering which is done to reduce noise and distortion in the image. The next



**Fig. 2** Flow diagram of license plate recognition

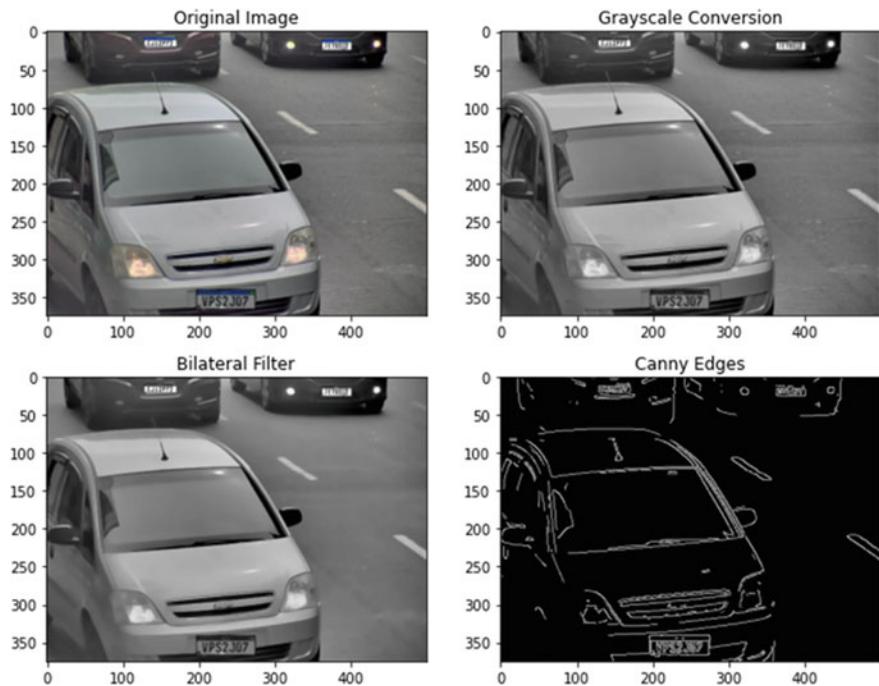
step in this preprocessing step is to detect the edges, which is done by the canny edge detector. Figure 3 shows all the outputs of all the steps in preprocessing. Similar preprocessing step is used in check legal amount detection also.

### 2.3.2 Check Amount and License Plate Detection

After the edge detection process, all contours are discovered from the edge detected image, and any contours with an area of less than a fixed threshold are removed, leaving only the remaining contours to be processed further. Each contour is approximated to form a polygon, and if a contour has four sides, it is expected to be a number plate (license plate) or amount text box (check), and the contours are drawn in the original image. The output of the above-mentioned process is presented in Fig. 4.

### 2.3.3 Perspective Transformation

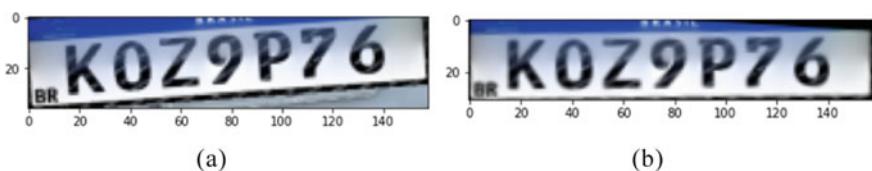
Depending on the angle of the object and camera planes, the license plate projection on the image looks varied. The projection transformation is used to carry out the rotation so that the slanted license plate is de-slanted. Figure 5 represents the slanted and de-slanted images of the license plates. This transformation is not applied on the bank check amount as the bank checks are scanned in a well-controlled environment, whereas the license plate images are captured in an unconstrained environment.



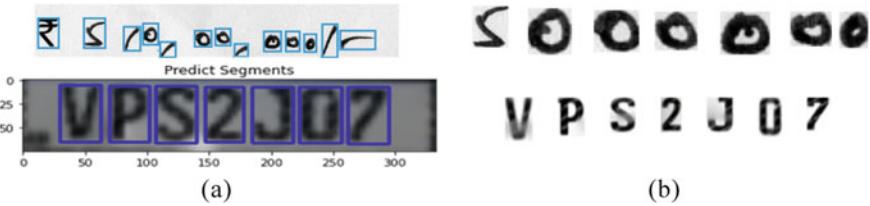
**Fig. 3** Sample images of every preprocessing step for a license plate image



**Fig. 4** License plate and bank check amount detected image



**Fig. 5** **a** Slanted license plate. **b** De-slanted license plate



**Fig. 6** **a** Contour detected image. **b** Segmented individual characters

### 2.3.4 Character Segmentation

The license plate and the bank check amount are separated into its constituent pieces in order to acquire the characters independently during plate character segmentation. The next step is to cut the plate characters. It is done by finding contours for the characters in the license plate image. Most of the check amounts are segmented correctly except for five check amount images as they were having a running letter style. Those images were cropped manually as the objective of this work is to verify the accuracy of the recognition.

The special characters such as comma (,), slash (/), rupee symbol ( $\text{₹}$ ) are ignored for further processing even though they are detected as characters. Figure 6 shows the contour detection and the individual segmented characters.

We have used the unified feature descriptor [13] for the handcrafted features. In addition to the proposed descriptor, we have added the moment feature also along with those features for the feature extraction module. We have included the Regular, Central, and normalized moments. Then, the extracted features are fed into SVM classifier with linear kernel for classification purpose. SVM is a firm supervised machine learning algorithm, and it is used by researchers for many classification tasks such as disease classification [14], vehicle recognition [15], generic object recognition [16], Alzheimer's disease classification [17], and hyper-spectral image classification [18]. To verify the potential of deep learning, we have used the DIGI-Net [19] architecture and slightly modified the output layer with ten classes (0–9) for check recognition and 36 classes (0–9, A–Z) for license plate recognition.

## 3 Experimental Results and Discussions

For training, we have used the CVL single-digit dataset [20] for check legal amount recognition and 36 classes (A–Z and 0–9) of natural image dataset of Chars74K dataset [5] for the car license plate recognition problem. The accuracy of the SVM classifier and the DIGI-Net architecture for the check and license plate recognition is given from Tables 1 and 2. The confusion matrix for these classifications is denoted in Fig. 7.

**Table 1** Accuracy of unified features using SVM-based model

Application	Unified features	Accuracy (%)
Legal check amount recognition	HOG + Zoning	83.72
	HOG + Zoning + Moments	74.18
License plate recognition	HOG + Zoning	82.61
	HOG + Zoning + Moments	85.80

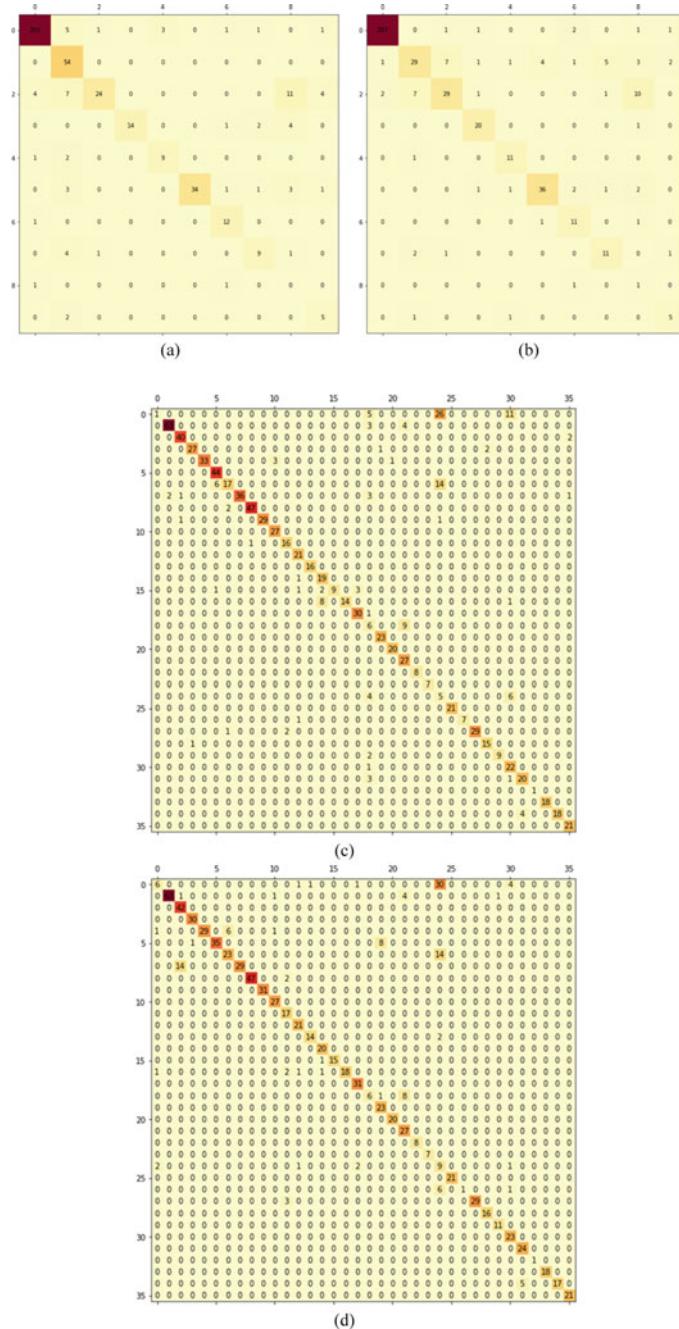
**Table 2** Accuracy of unified DIGI-Net CNN-based model

Application	Input size	Epochs			
		25	50	75	100
Legal check amount recognition	32 × 32	77.90	84.18	87.90	85.34
	64 × 64	87.90	89.30	86.27	85.58
License plate recognition	32 × 32	77.77	75.57	75.90	79.64
	64 × 64	75.57	82.94	82.39	84.26

From the recognition of check amount recognition, we can infer that even though the distribution of the test dataset may be different from the distribution of the trained CVL dataset, the performance of both unified features with SVM and the DIGI-Net CNN appears to be satisfactory when addressing handwritten character recognition issues. The performance of the  $32 \times 32$  sized DIGI-Net model is superior to that of the unified SVM model, whereas the DIGI-Net architecture that uses  $64 \times 64$  as the input size is still superior. It would be preferable if the legal amount could be validated by utilizing the value of the amount as it is written in English; however, doing so would involve additional processing. Despite the fact that there is still a significant amount of opportunity for advancement in this automated check processing, this study can be seen as a stepping stone toward the outcome that is desired.

According to the confusion matrix of license plate recognition, the number 0 is most commonly recognized as the letter O in both handcrafted and machine-crafted learned models. There are more instances of confusion between the digits 0 and O in DIGI-Net than there are in the unified feature extraction model, which has more instances of confusion between 0 and O but less instances of confusion with U (may be the effect of outer and inner skeleton zoning). Because of these ambiguities, the accuracy of DIGI-Net and unified features with SVM is both reduced to 84.26% and 85.80%, respectively.

The DIGI-Net model improves performance by between 5 and 7%, which means that the size of the image also plays a significant role in the process of natural image character recognition. In this particular application, the performance of the unified feature extraction-based SVM model is somewhat superior to that of the DIGI-Net model. It is also to be noted that even though the usage of moment features tends to be a failure in the handwritten digit recognition, it serves better during the natural image character recognition. These unified models (both handcrafted and machine-crafted



**Fig. 7** Confusion matrix: **a** Check recognition—SVM, **b** check recognition—DIGI-Net, **c** license plate recognition—SVM, and **d** License plate recognition—DIGI-Net

model) do a comparatively satisfactory results excluding the fundamental disparity that exists between the digits 0 and O, V and Y, I and L, and 6 and O. These are common problems with the recognition of English characters, and they can be fixed by using strategies based on context.

## 4 Conclusion

In this research work, we have verified the usage of a unified character recognition system by recognizing the bank check amounts and the license plate number from a car image using traditional and deep learning method. We have obtained a maximum accuracy of 89.30% for bank check recognition and 85.80% for the license plate recognition. It is to be noted that the proposed deep learning system has provided promising results for check recognition systems. But, it is also to be noted that the handwritten recognition has many research capabilities as there is still room for improvement. It is also observed that the license plate recognition system has to be improved as the images obtained from the license plate are not of high quality and also it paves way for the exploration of new methodologies to be researched in the area of natural image character recognition. Application of these proposed systems for the real-world scenario is still a real challenge.

## References

1. Downton AC, Leedham CG (1990) Preprocessing and presorting of envelope images for automatic sorting using OCR. *Pattern Recogn* 23(3–4):347–362. ISSN 0031-3203. [https://doi.org/10.1016/0031-3203\(90\)90022-D](https://doi.org/10.1016/0031-3203(90)90022-D)
2. Li L, Zhang LL, SU J-F (2012) Handwritten character recognition via direction string and nearest neighbor matching. *J China Univ Posts Telecommun* 19(Suppl 2):160–196. ISSN 1005-8885. [https://doi.org/10.1016/S1005-8885\(11\)60427-5](https://doi.org/10.1016/S1005-8885(11)60427-5)
3. Manjunath Aradhy VN, Hemantha Kumar G, Noushath S (2008) Multilingual OCR system for South Indian scripts and English documents: an approach based on Fourier transform and principal component analysis. *Eng Appl Artif Intell* 21(4):658–668. ISSN 0952-1976. <https://doi.org/10.1016/j.engappai.2007.05.009>
4. Ali M, Foroosh H (2016) A holistic method to recognize characters in natural scenes. In: Proceedings of the 11th Joint conference on computer vision, imaging and computer graphics theory and applications—Volume 4: VISAPP, (VISIGRAPP 2016), pp 449–457. ISBN 978-989-758-175-5; ISSN 2184-4321. <https://doi.org/10.5220/0005787904490457>
5. De Campos TE, Babu BR, Varma M (2009) Character recognition in natural images. In: Proceedings of the International conference on computer vision theory and applications (VISAPP). Lisbon, Portugal, vol 2, pp 273–280. <https://doi.org/10.5220/0001770102730280>
6. Guillevic D, Suen CY (1998) Recognition of legal amounts on bank checks. *Pattern Anal Appl* 1:28–41. <https://doi.org/10.1007/BF01238024>
7. Ghosh R, Panda C, Kumar P (2018) Handwritten text recognition in bank checks. In: 2018 Conference on information and communication technology (CICT), pp 1–6. <https://doi.org/10.1109/INFOCOMTECH.2018.8722420>

8. Palacios R, Gupta A, Wang PS (2004) Handwritten bank check recognition of courtesy amounts. *Int J Image Graph* 4(02):203–222. <https://doi.org/10.1142/S0219467804001373>
9. Ozbay S, Ercelebi E (2007) Automatic vehicle identification by plate recognition. *Int J Electron Commun Sci* 9. <https://doi.org/10.5281/zenodo.1331865>
10. Kim T-G, Yun B-J, Kim T-H, Lee J-Y, Park K-H, Jeong Y, Kim HD (2021) Recognition of vehicle license plates based on image processing. *Appl Sci* 11:6292. <https://doi.org/10.3390/app11146292>
11. Dansena P, Bag S, Pal R (2017) Differentiating pen inks in handwritten bank cheques using multi-layer perceptron. In: Shankar B, Ghosh K, Mandal D, Ray S, Zhang D, Pal S (eds) *Pattern recognition and machine intelligence. PReMI 2017. Lecture notes in computer science*, vol 10597. Springer, Cham. [https://doi.org/10.1007/978-3-319-69900-4\\_83](https://doi.org/10.1007/978-3-319-69900-4_83)
12. Silvano GVT, Ivanovitch S, Ribeiro VCT, Greati VR, Bezerra A, Endo PT, Lynn T (2020) Artificial Mercosur license plates dataset. *Data Brief* 33:106554. ISSN 2352-3409. <https://doi.org/10.1016/j.dib.2020.106554>
13. Arun M, Arivazhagan S (2022) A unified feature descriptor for generic character recognition based on zoning and histogram of gradients. *Neural Comput Appl.* <https://doi.org/10.1007/s00521-022-07110-x>
14. Ahila Priyadharshini R, Arivazhagan S, Arun M et al (2019) Maize leaf disease classification using deep convolutional neural networks. *Neural Comput Appl* 31:8887–8895. <https://doi.org/10.1007/s00521-019-04228-3>
15. Priyadharshini RA, Arivazhagan S, Sangeetha L (2014) Vehicle recognition based on Gabor and Log-Gabor transforms. In: 2014 IEEE international conference on advanced communications, control and computing technologies, pp 1268–1272. <https://doi.org/10.1109/ICACCCT.2014.7019303>
16. Ahila Priyadharshini R, Arivazhagan S (2014) A quaternionic wavelet transform-based approach for object recognition. *Defence Sci J* 64(4)
17. Divya R, Shantha Selva Kumari R, The Alzheimer's Disease Neuroimaging Initiative (2021) Genetic algorithm with logistic regression feature selection for Alzheimer's disease classification. *Neural Comput Appl* 33:8435–8444. <https://doi.org/10.1007/s00521-020-05596-x>
18. Kavitha K, Arivazhagan S, Kayalvizhi N (2010) Wavelet based spatial—spectral hyperspectral image classification technique using Support Vector Machines. In: Second international conference on computing, communication and networking technologies, Karur, India, pp 1–6. <https://doi.org/10.1109/ICCCNT.2010.5591760>
19. Madakannu A, Selvaraj A (2020) DIGI-Net: a deep convolutional neural network for multi-format digit recognition. *Neural Comput Appl* 32:11373–11383. <https://doi.org/10.1007/s00521-019-04632-9>
20. Diem M, Fiel S, Garz A, Keglevic M, Kleber F, Sablatnig R (2013) ICDAR 2013 competition on handwritten digit recognition (HDRC 2013). In: Proceedings of the 12th international conference on document analysis and recognition (ICDAR) 2013, pp 1454–1459

# **$\alpha$ and $\beta$ -Testing of an Epileptic Seizure Detection Algorithm on Pre-ictal, Ictal, and Inter-ictal Part of EEG Signal**



**Khakon Das** and **Ashish Khare**

**Abstract** Epilepsy is the main neurological disorder, which leads to human brain working incorrectly. Hundreds of thousands of people are stricken by this sickness. The random nature of seizure occurrence makes it vulnerable. The present work is direction of building any such gadget, which can supply a caution message to take a safety degree earlier than the seizure happens. For this, we used epileptic EEG signals. Following a thorough assessment, we unveiled several distinctive characteristics exhibited by epileptic EEG signals, including (i) background disruption, (ii) well-defined spike segments, (iii) electro cerebral negativity, and (iv) electrical field. We have designed an algorithmic model considering all of the above mentioned characteristics of epilepsy. By issuing an advance warning message, this model can provide information about the occurrence of predisposed seizure waveform from a specific part of the EEG signal. This will offer enormous help to the doctors in detecting the seizure from the pre-ictal, ictal, and inter-ictal parts of the EEG signal. This automated system provides a good handy solution to doctors and healthcare professionals. This system has been tested on various datasets and achieved 91.07% sensitivity, 97.37% specificity, and 99.03% positive predictive value and 92.67% accuracy for primary dataset, and in case of the secondary dataset, the proposed method achieved 100%, 92%, and 100% accuracy for the ictal part, pre-ictal, and inter-ictal parts, respectively, for the considered datasets which is clinically acceptable.

**Keywords** Electroencephalogram · Epilepsy · Seizure · Pre-ictal · Ictal ·  $\alpha$ -testing

---

K. Das · A. Khare (✉)

Department of Electronics and Communication, University of Allahabad, Allahabad, Uttar Pradesh 211002, India

e-mail: [khare@allduniv.ac.in](mailto:khare@allduniv.ac.in)

K. Das

e-mail: [khakon.phd2021@allduniv.ac.in](mailto:khakon.phd2021@allduniv.ac.in)

## 1 Introduction

Epileptic seizures are the most common chronic neurological non-communicable disorder, affecting around 50 million people worldwide [1, 2]. The impact and incidence of epilepsy are more in infants than adults. There will be a significant burden on patients with delay in pathological diagnosis of epilepsy. The chances of decreased quality of life will be more with inaccurate epileptic seizure management. The above mentioned clauses raised a great need of early detection and diagnosis of epileptic seizure. Accurate epilepsy detection optimizes patient care. A seizure is a signal event characterized by sudden abnormality of brain's abnormal electrical activity. An epileptic seizure is categorized in a several forms, and recurrent seizures are one of them. Sudden abnormalities in brain's electrical signals are observed in a regular time intervals [3, 4]. An epileptic seizure disturbs functionality of several organs of a patient, and its potential implications include cognizance loss or a whole-frame convulsion. It is tuff to recognize mild seizures because of its sloppy nature; in turn, the risk of permanent damage increases exponentially. Implications of epileptic study on world's population revealed that 4–5% of the world's population is affected by epilepsy, and 25% of out of them are not getting proper treatment with lack of proper assistance [5]. An automatic interpretation of epilepsy from EEG signals has been attempted formerly by the usage of artificial neural networks [6–8]. Several alternative methodologies, such as hyper-clustering in lengthy EEG records, have also surfaced, demonstrating a concordance rate of 91% with visual interpretation [9]. At the same time, early algorithms had reported low sensitivity and specificity in prediction of epileptic seizure from the pre-ictal part of EEG signal. Recent methods have achieved a sensitivity and specificity of 94% [10, 11]. Das et al. [12–14] proposed a classification model that pre-detects epileptic seizures from the pre-ictal part of EEG signals. Their model was tested on data of several normal people's EEG and 150 epileptic patients EEG and successfully detected epileptic seizures in advance with 92.66% accuracy. An innovative EEG signal analysis for seizure detection has been proposed by Gandhi et al. [15, 16] using probabilistic neural network, statistical features, and discrete wavelet transform-based seizure detection. Although experimental results of their model have shown 100% accuracy, their seizure detection has been carried only from ictal part only. Hence, their model has shown less than 20% accuracy during the presence of poly-spike, rhythmic wave, and low-amplitude like properties in EEG signal. Hence, there is a strong need to develop such systems that can cope up with the mentioned loop holes. The time required for manual diagnosis of EEG by the healthcare provider can be greatly reduced with an automated seizure detection system. This type of system can serve as an electrical stimulator to provide pre detection of seizures that can greatly aid in improvising the epileptic patient's life expectancy.

The following are the contributions of the proposed model for detecting epileptic seizures.

1. From the pre-ictal portion of the EEG signal, the model is capable of detecting epileptic seizures.

2. The model can successfully work on low-amplitude EEG signals.
3. The model can successfully detect complex epileptic seizure waveforms.
4. The model is also capable to deliver a warning signal.

Rest of the paper is organized as follows: Sect. 2 presents implementation details of tools and specifications. Section 3 demonstrates the proposed methodology. Section 4 presents the experimental results and analysis, while Sect. 5 serves as the conclusion of the paper.

## 2 Implementation Details

This section is subdivided into three subsections, first one is about seizure detection sensor technology, and others two are tools and dataset.

### 2.1 *Seizure Detection Sensor Technology*

A brain signal can be detected through different sensors such as unimodal detection devices and multimodal detection devices. Under unimodal, sensor types may be electroencephalogram (EEG), surface electromyography (sEMG), and electrodermal activity (EDA). A multimodal sensor is a combination of unimodal sensors to improvise the signal accuracy. EEG has been regarded as the gold standard for detecting seizures. Hence, in our experimentation, we use EEG. In this method, the electrodes, which are placed along the scalp, transmit the information, from the brain to a recording machine. A seizure trigger episode always characterized as a rapid spike in EEG signal. Hence, with the investigation of electroencephalogram signal, it is possible to detect the chances of seizure episodes in past and future.

### 2.2 *Tools*

The programming code of the proposed model has been developed in MATLAB R2017a (64 bit) and windows 10 (64 bit) hardware environment with system configuration of 8 GB RAM and the processor technology of Intel Core i7.

### 2.3 *Dataset*

We use two datasets, one is the primary dataset, and the other one is the secondary dataset. Both datasets will be discussed briefly below.

### 2.3.1 Primary Dataset

The dataset was collected from the Department of Neurology at N.R.S.M.C. & Hospital in Kolkata, West Bengal. A total of 150 patients suffering with neurological disorder has been undergone EEG test clinical evaluation with their willingness consent to utilize their pathological data for research purpose. Their EEG recordings were done using 16-channels RMS Brain-View equipment of 256 Hz sampling frequency, HFF-70 Hz, LFF-0.1 Hz. The EEG recordings were visually inspected, and segments without epileptic-form discharge were exported to excel file format. Each file contains 10 s of EEG pre-ictal signal segment [14].

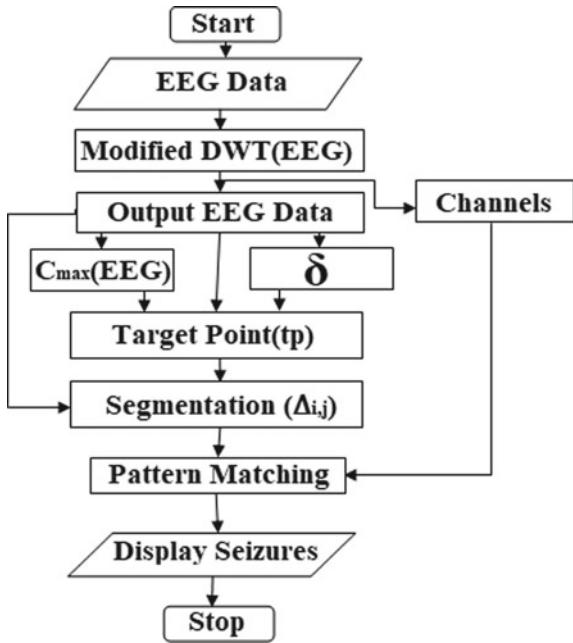
### 2.3.2 Secondary Dataset

This dataset has been developed as part of a research work that has been carried in collaboration of Neuro computing Laboratory, IIT Delhi and Neurology and Sleep Centre, Hauz Khas. The datasets they have consist of segmented time series EEG recordings from ten patients who have been diagnosed with epilepsy. The Grass-Tele-Factor-Comet (AS40) Amplification System was used to acquire the data at a 200 Hz sampling rate. The scalp EEG electrodes were placed in accordance with the 10–20 electrode-placement-systems during signal acquisition. Band pass filters were used to filter the signals between 0.5 and 70 Hz, and they were subsequently segmented into pre-ictal, inter-ictal, and ictal stages. Each file contains 1024 samples of one EEG time series data lasting for 5.12 s [15, 16].

## 3 Design Methodology of the Proposed Work

The proposed algorithm for automated epilepsy detection has been developed based on analysis of EEG signals, which are widely used in various cerebral diagnostics. An increasing range of research has been done using this technology. We have worked with information obtained from EEG. Exploring the same, we applied segmentation and feature extraction to improve the further classification of the EEG signal [17–22]. The concept of lifting scheme was first introduced by Sweldens in 1996 [23]. This scheme relies on in-place calculation [13, 24] and effectively eliminates the requirement for excessive memory. We used this concept in our proposed model. Due to simple mathematical calculation and wide applications in medical signal and image processing, we used the Haar wavelet as mother wavelet [25–27] for denoising of EEG signals. By utilizing the nearest neighbor method, the suggested algorithm examines the discharge field following the identification of sudden alterations in the amplitude, power, and energy of the EEG signal. The slope and direction of changes are incorporated in the algorithm. We used MATLAB as our platform to work with the EEG data.

**Fig. 1** Flowchart of proposed work



### 3.1 Flowchart of Proposed Work

A flowchart representation of proposed work has been shown in Fig. 1.

### 3.2 Algorithm(s)

In this subsection, we will provide a detailed discussion of the given flowchart showing in Fig. 1, focusing on each part separately.

#### (a) Algorithm for Calculating Local $m$ Maxima

$$C_{\text{max}} = C_{\text{max}} = (\text{Max } (\text{EEG}(a, b)) < C_{\text{max}}); \quad (1)$$

Using Eq. 1, we calculate the local maxima value, here refer as ‘ $C_{\text{max}}$ ’. It is a point of suspect where the seizure may be located. The current maxima position becomes epileptic seizure location or not that we decided through step-by-step execution of Eqs. 2, 3, and 4. In Eq. 1, letter ‘ $a$ ’ defines the current channel number which lies between 1 and ‘ $N$ ’; here, the value of ‘ $N$ ’ is 16. And the letter ‘ $b$ ’ defines the current

time index of a particular channel. The minimum and maximum value of the ‘ $b$ ’ are 1 and ‘CL’, respectively, where ‘CL’ represents the length of the EEG signals.

(b) *Algorithm for Calculating Threshold Value*

$$\partial \left( \frac{\text{EEG}}{\text{Cmax(index)}} \right) = \sum_{\substack{a \geq 1 \\ b = \text{Cmax(index)}}}^N \text{EEG}(a, b) \quad (2)$$

where ‘ $\partial$ ’ (calculate using Eq. 2) refers to a soft threshold value at a particular position which is defined by ‘Cmax(index)’ of EEG signal (calculate using Eq. 1). Here ‘ $a$ ’ denotes the EEG channels, and ‘ $N$ ’ defines the total number of EEG channels.

(c) *Target Point Selection*

We use a target point selection algorithm as an optimization technique to identify local maxima or minima at specific points in a waveform (EEG signal) across ‘ $N$ ’ identical channels.

Let, a straight line equation is

$$Ax + By + C = 0; \quad (3)$$

Assuming that the straight line equation is  $Ax + By + C = 0$ , the line is said to contain the maximum number of local maxima points (as calculated from Eq. 1). At the index with the highest number of current maxima points, the line will intersect all EEG channels vertically.

(d) *Segmentation*

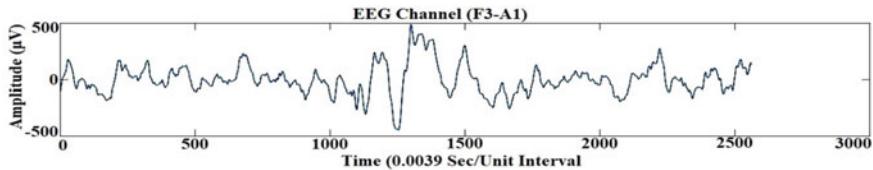
Each and every distinct segment ( $\Delta_{a,b}$ ) is selected (using Eq. 4), which is a strict subset of the all EEG segments frame of  $a^{th}$  channel.

$$\text{Segment}(\Delta_{a,b}) \subset \text{EEG}_{a^{th} \text{ Channel and } b^{th} \text{ index}} \quad (4)$$

where  $a = 1, 2, 3, \dots, N$ . and  $b = 1, 2, 3, \dots, CL$ . [Note: N and CL have maximum value as same as define in the Eq. 1]

(e) *Pattern Matching*

Split all segments ( $\Delta_{a,b}$ ) (obtained from the Eq. 4) into two equal-length sub-segments from the center point, with one sub-segments considered as background and the other as foreground. And compare both to investigate the disturbed background. Our main focus is to calculate and compare various features of background and foreground in frequency and energy stage.



**Fig. 2** Sample EEG data of frontal lobe channel 3(F3)

## 4 Output and Result Analysis

This segment is further subdivided into three subsections, namely input dataset and output analysis of primary and secondary datasets.

### 4.1 Input Datasets

We are using two datasets, which are described below.

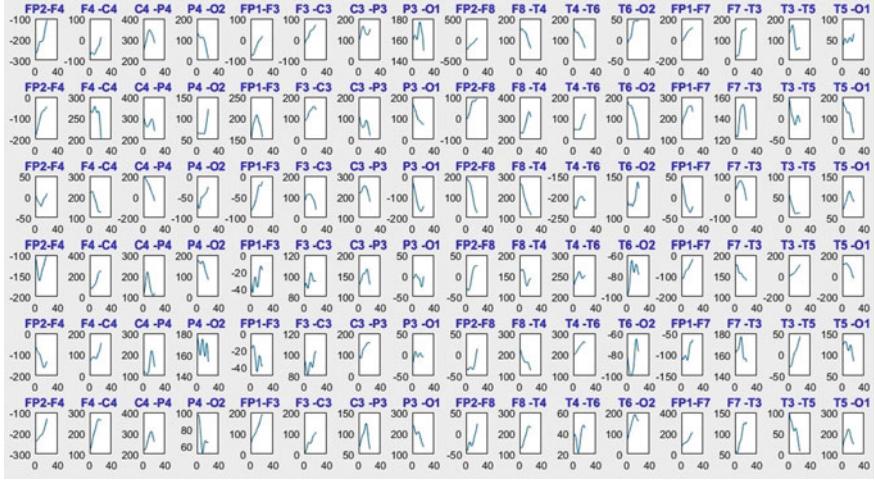
**Primary Dataset:** For experimentation, we have taken the denoised EEG recording through 16 individual channels; out of those, the EEG data of frontal lobe channel-3 (F3) is shown in Fig. 2. The duration of the window of the EEG is 10 s. The EEG records obtained from all 16 channels are taken into consideration when the algorithm is applied.

**Secondary Dataset:** The secondary dataset's EEG signals were recorded at a sampling rate of 200 Hz, and band pass filters were used to filter the signals between 0.5 and 70 Hz. The signals were then segmented into pre-ictal, inter-ictal, and ictal stages, with each file comprising a one EEG time series data of 1024 samples and lasting for duration of 5.12 s.

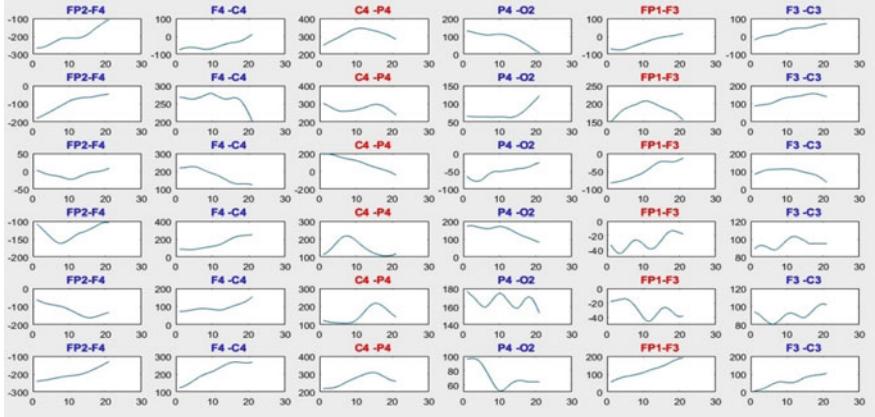
### 4.2 Output Analysis of Primary Dataset

The system-generated intermediate output after applying the target point set and segmentation operation of the proposed algorithm is shown in Fig. 2. Figure 3 contains all those utmost probable segments, where the seizure may present. Here, each row represents a target point, and each column represents a channel starting from channel 1 to channel 16.

Figure 4 shows the targeted channels, where seizure is detected which occurred in central lobe channel-4 (C4-P4) and fronto-parietal lobe channel-1 (FP1-P3).



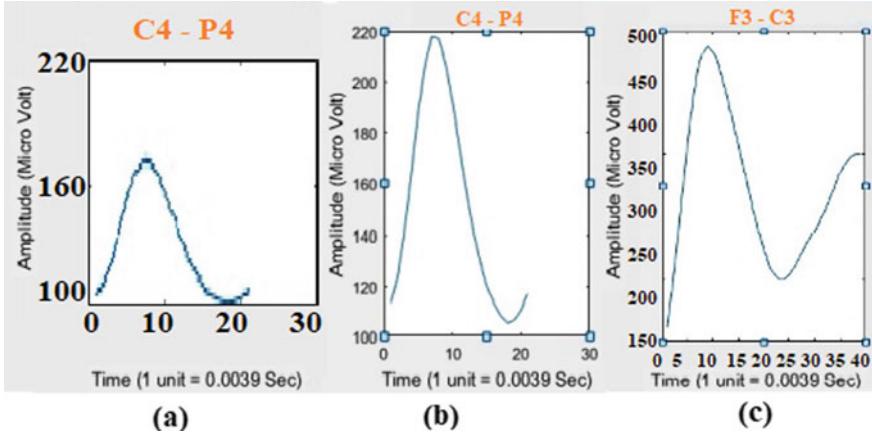
**Fig. 3** Segments with probable seizure



**Fig. 4** Target channels where seizure is detected

#### 4.2.1 Seizure Spike

The duration of spikes and sharp waves is crucial in the study of epilepsy. Spikes typically last for less than 70 ms, while sharp waves, which have a duration between 70 and 200 ms, are also important indicators of epileptic activity [14]. Figure 5 shows the segment with the individual channel name. We filtered all affected channels using the domain matcher algorithm. Figure 5b shows a magnified description of the central lobe channel (C4) shown in Fig. 5a, where a typical sharp spike is observed. The duration of the spike is  $18 \times 0.0039$  s = 0.0702 s, which justifies phrasing it as a sharp spike. Figure 5c shows a typical spike and sharp wave, obtained from the EEG



**Fig. 5** Snapshot of typical seizure; **a** spike, **b** sharp spike, and **c** sharp spike and wave

record of another patient and applying the same algorithm to it. The observation made from Fig. 5b and c is that the waves require a very short duration to meet the criteria for epilepsy detection.

#### 4.3 Output Analysis of Secondary Dataset

A secondary dataset obtained from IIT Delhi has been used for  $\beta$ -testing purpose. The said dataset contains 150 files. This dataset is divided into three parts, namely ictal, per-ictal, and inter-ictal. All three parts contain 50 files for each type. Now, we run our algorithm using this dataset. The proposed method successfully detected seizure from the ictal part, and additionally, the proposed method also successfully predicts seizure waveform from the pre-ictal part of this dataset. In Fig. 6, we have shown all types of seizure waveforms detected from the ictal part by the proposed methods. Firstly, we categorized those waveforms into simple and complex waveforms, where simple waveforms are further classified into a sharp spike and thick spike waveforms.

To do this, we consider time, phase, and amplitude features. The time duration of a sharp spike and spike is less than 70 ms, sharp wave duration is between 70 and 200 ms, and slow wave and spike-and-wave duration are about 350 ms [28]. The complex waveform is a combination of two different categories like spike and wave, sharp spike and wave, spike and slow wave, and spike and sharp wave. Figure 7 is shown the same detection from the inter-ictal part of the secondary dataset, and Fig. 8 contains the detected waveforms from the pre-ictal part of the secondary dataset. All waveforms are labeled by proper subcategories' waveform patterns that are clinically known. The duration of the spike in Fig. 6.1 is  $15 \times 0.005 \text{ s} = 0.075 \text{ s}$ , which justifies it as a sharp spike.

Fig: 6 Secondary Data Set: Some seizure patterns discovered from the given ictal Dataset.

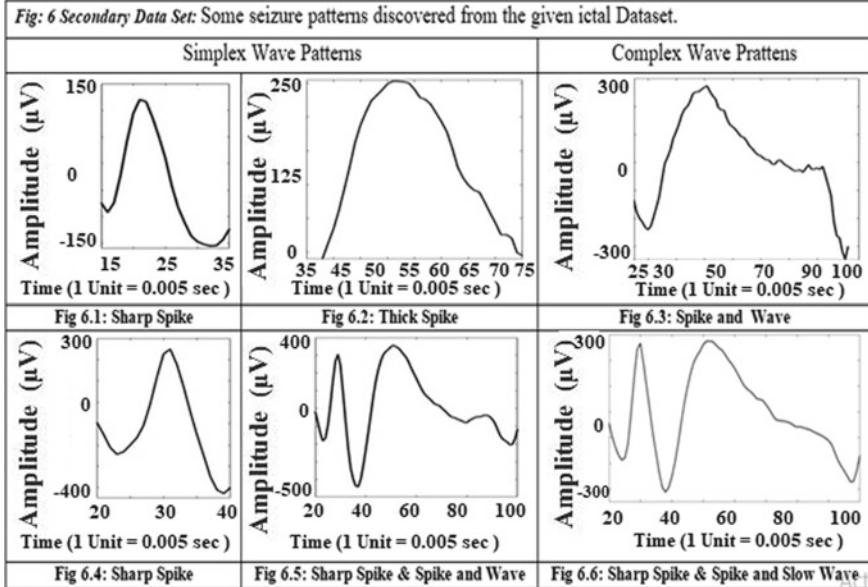


Fig. 6 Some seizure patterns discovered from the given ictal part (secondary dataset)

Fig 7: Secondary Data Set: Some seizure patterns discovered from the given Inter-ictal Dataset.

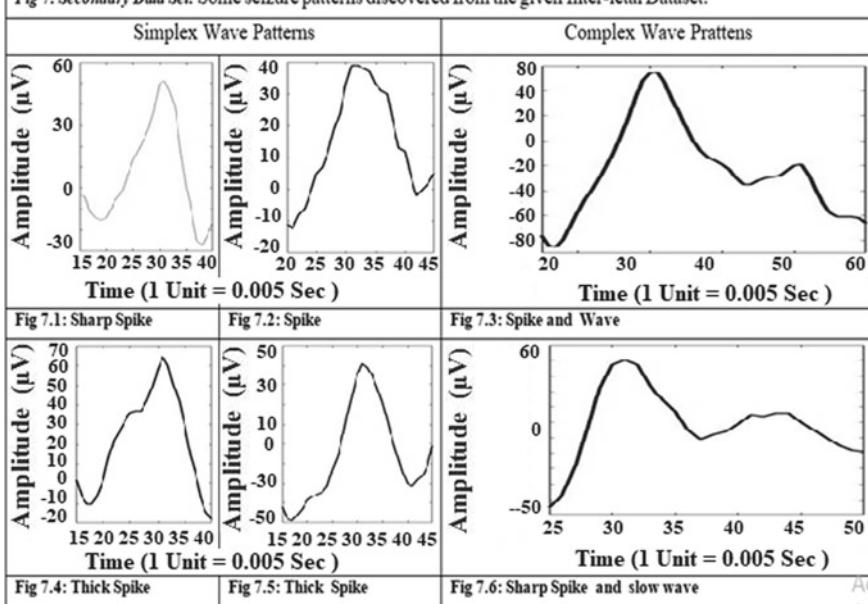
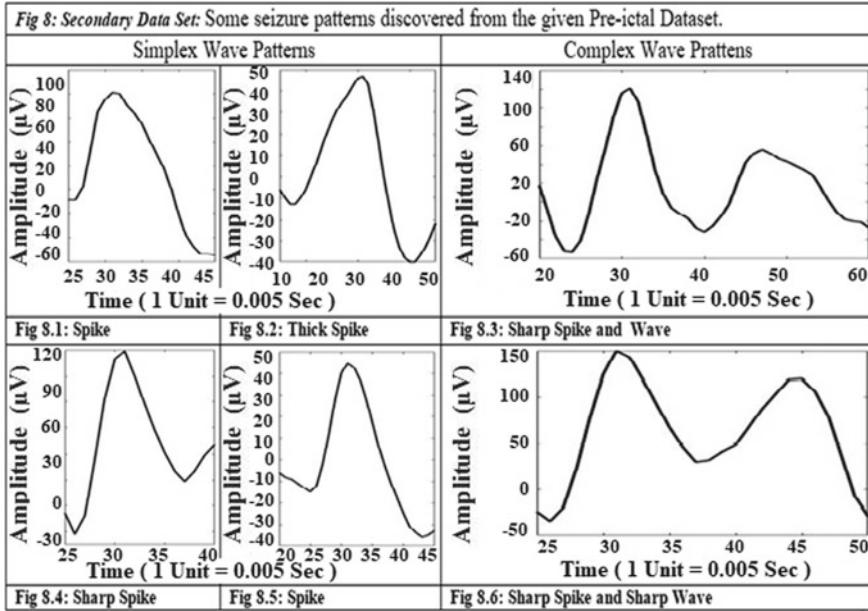


Fig. 7 Some seizure patterns discovered from the given inter-ictal part (secondary dataset)



**Fig. 8** Some seizure patterns discovered from the given pre-ictal part (secondary dataset)

#### 4.4 Discussion

After applying the proposed algorithm, we acquired incredibly affirmative results for advanced seizure prediction and analysis. We have tested our algorithm on 150 subjects of the primary dataset. We achieved 91.07% sensitivity, 97.37% specificity, 99.03% positive predictive value, and 92.67% accuracy (showing in Table 1). If we consider the number of samples, used in our experiment that it is relatively greater in number than the experiments done in [29, 30]. Our algorithm gives a better overall performance than that of the performance achieved by the methods [29, 30]. In case of the secondary dataset, the proposed method achieved 100% accuracy considering the ictal part of the EEG signal which contains 50 files containing on-set seizure. The proposed method can also detect seizure from the pre-ictal (50 files) and interictal (50 files) parts of secondary dataset with 92% and 100% accuracy, respectively (showing in Table 1). The methods of [15, 16] are only capable to detect seizures from the ictal part. The model mentioned in [31] is also capable to detect seizures from the per-ictal part only. But the proposed method outperformed all mentioned methods by detecting seizures from all three part of the EEG signals.

**Table 1** Display the sensitivity and specificity metrics for the utilized datasets

Calculate sensitivity, specificity, accuracy, <i>PPV</i> , and <i>NPV</i>						
Details	Data samples	Sensitivity TP/(TP + FN)	Specificity TN/(FP + TN)	Accuracy (TP + TN)/(P + N)	Positive predict value TP/(TP + FP)	Negative predict value TN/(TN + FN)
Primary dataset		150 data samples				
True positive	102	91.07%	97.37%	92.67%	99.03%	78.72%
False negative	10					
False positive	01					
True negative	37					
Secondary dataset		Pre-ictal part (50 data samples)				
True positive	46	92.00%	–	92.00%	100%	0.0%
False negative	4					
False positive	0					
True negative	0					
Secondary dataset		ictal and inter-ictal part (50 + 50 = 100 data samples)				
True positive	100	100%	–	100%	100%	–
False negative	0					
False positive	0					
True negative	0					

*Note* True positive = **TP**, false negative = **FN**, false positive = **FP**, true negative = **TN**

## 5 Conclusion and Future Scope of Work

We have proposed an algorithm for detecting epileptic-form discharges that demonstrates significant agreement with the visual interpretation of neurology experts. As the number of channels (such as 16-ch, 32-ch, 128-ch, and so on) in EEG devices continues to increase, it has become challenging for healthcare professionals to continuously monitor EEG signals across all channels and visually inspect them

simultaneously for an extended period. The proposed algorithm will provide a solution in all cases where seizures are present in the EEG by considering all channels. Also where there is no seizure witnessed by visual interpretation, the proposed algorithm detects probable seizure form in it and produces an alert to capture the neurologists' attention. The proposed algorithm will assist neurologists to locate the presence of a seizure from ictal, pre-ictal, and inter-ictal parts of epileptic EEG recording, thus help in saving the precious life of human beings.

In our future research, we plan to enhance the robustness of the algorithm by employing a larger sample size. However, a significant limitation of the proposed work is that the cases are not categorized as focal or generalized discharges, which is currently under investigation.

**Acknowledgements** Throughout this research work, valuable comments were provided by Dr. Shankar Prasad Saha (Neurologist) of N.R.S. M. C. & H in Kolkata, West Bengal, India, for which we express our sincere gratitude.

## References

1. World Health Organization (n.d.) Epilepsy. World Health Organization. Last Accessed: 30 Jan 2023 from <https://www.who.int/news-room/fact-sheets/detail/epilepsy>
2. Chang BS, Lowenstein DH (2003) Epilepsy. *N Engl J Med* 349:1257–1266
3. Dreifuss FE (1981) Proposal for revised clinical and electroencephalographic classification of epileptic seizures. *Epilepsia* 22:489–501
4. Hanscomb A, Hughes L (1995) Epilepsy (ward lock family health guides). National Society for Epilepsy. Publisher: Ward Lock Ltd., London, UK, London, p 80. Type: Non-serial independent book, ISBN-13: 978-0706374049
5. Lehnertz K et al (2003) Seizure prediction by nonlinear EEG analysis. *IEEE Eng Med Biol Mag* 22(1):57–63
6. Webber WRS (1994) *Neuron* 91:194–204
7. Khan YU, Farooq O, Sharma P (2012) Automatic detection of seizure onset in paediatric EEG. *Int J Embed Syst Appl* 2(3):81–89
8. Faust O, Rajendra Acharya U, Adeli H, Adeli A (2015) Wavelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis. *Seizure* 26:56–64
9. Kural MA, Duez L, Sejer Hansen V, Larsson PG, Rampp S, Schulz R, Tankisi H, Wennberg R, Bibby BM, Scherg M, Beniczky S (2020) Criteria for defining inter-ictal epileptiform discharges in EEG: a clinical validation study. *Neurology* 94(20):e2139–e2147
10. Pfammatter JA, Maganti RK, Jones MV (2019) An automated, machine learning–based detection algorithm for spike-wave discharges (SWDS) in a mouse model of absence epilepsy. *Epilepsia Open* 53(7):1196–1204
11. Frost JD Jr (1985) Automatic recognition and characterization of epileptiform discharges in the human EEG. *J Clin Neurophysiol* 2(3):231–249
12. Mallat SG (1989) A theory for multi-resolution signal decomposition: the wavelet representation. *Pattern Anal Mach Intell IEEE Trans* 11(7):674–693
13. Khakon D, Mausumi M, Punit S, Minakshi B (2018) Early started hybrid denoising technique for medical images. In: Recent trends in signal and image processing. Advances in intelligent systems and computing, vol 727, pp 131–140. Springer, Singapore
14. Das K, Daschakladar D, Roy PP, Chatterjee A, Saha SP (2020) Epileptic seizure prediction by the detection of seizure waveform from the pre-ictal phase of EEG signal. *Biomed Sig Process Control* 57:101720

15. Swami P, Gandhi T, Panigrahi BK, Tripathi M, Anand S (2016) A novel robust diagnostic model to detect seizures in electroencephalography. *Expert Syst Appl* 56:116–130
16. Gandhi TK, Chakraborty P, Roy GG, Panigrahi BK (2012) Discrete harmony search based expert model for epileptic seizure detection in electroencephalography. *Expert Syst Appl* 39(4):4055–4062
17. Ararabi A, Grebe R, Wallois F (2007) A multistage knowledge-based system for EEG seizure detection in newborn infants. *Clin Neurophysiol* 118(12):2781–2797
18. Greene BR, Faul S, Marnane WP, Lightbody G, Korotchikova I, Boylan GB (2008) A comparison of quantitative EEG features for neonatal seizure detection. *Clin Neurophysiol* 119(6):1248–1261
19. Temko A, Thomas E, Boylan G, Marnane W, Lightbody G (2009) An SVM-based system and its performance for detection of seizures in neonates. In: Annual International conference of the IEEE engineering in medicine and biology society. Institute of Electrical and Electronics Engineers, Minneapolis, MN, pp 2643–2646
20. Shoeb AH, Guttag JV (2010) Application of machine learning to epileptic seizure detection. In: ICML'10: Proceedings of the 27th International conference on machine learning, pp 975–982. <https://dl.acm.org/>. <https://doi.org/10.5555/3104322.3104446>
21. Liang SF, Wang HC, Chang WL (2010) Combination of EEG complexity and spectral analysis for epilepsy diagnosis and seizure detection. *EURASIP J Adv Sig Process* 2010:853434
22. Oweis RJ, Abdulhay EW (2011) Seizure classification in EEG signals utilizing Hilbert-Huang transform. *BioMedEng Online* 10(38):38
23. Daubechies I, Sweldens W (1998) Factoring wavelet transform into lifting steps. *J Fourier Anal Appl* 4(3):247–269
24. Koichi K, Koichi N, Shigeru T (2004) FPGA-based lifting wavelet processor for real-time signal detection. *Signal Process* 84(10):1931–1940
25. Stankovi RS, Falkowski BJ (2003) The Haar wavelet transform: its status and achievements. *Comput Electr Eng* 29(1): 25–44
26. Khare A, Tiwary US, Pedrycz W, Jeon M (2010) Multilevel adaptive thresholding and shrinkage technique for denoising using Daubechies complex wavelet transform. *Imaging Sci J* 58(6):340–358
27. Bommisetty RM, Khare A, Khare M, Palanisamy P (2022) Content based video retrieval using integration of Curvelet transform and simple linear iterative clustering. *Int J Image Graph* 22(2). Article id 2250018
28. Jaseja H et al (2012) EEG spike versus EEG sharp wave: differential clinical significance in epilepsy. *Epilepsy Behav* 25(1):137
29. Litt B et al (2001) Epileptic seizures may begin hours in advance of clinical onset: a report of five patients. *Neuron* 30:51–64
30. Ahammad N, Fathima T, Joseph P (2014) Detection of epileptic seizure event and onset using EEG. *Bio Med Res Int* 2014:7. Article ID 450573
31. Das K, Saha SP, Singh KK (2020) Detection of epileptiform seizure from pre-ictal part of epileptic EEG recording. In: Advances in systems analysis, software engineering, and high performance computing. IGI Global, pp 36–49

# SBGAN: Sequential Bengali Word Image Generation Model



Piyush Kanti Samanta, Arpita Dutta, and Samit Biswas

**Abstract** The rapid growth and usage of camera-based mobile devices have increased the number of camera-captured documents; these documents may be printed or handwritten. In a nation like India, multi-script recognition from document images is a critical step before choosing a suitable script-specific optical character recognition (OCR). The handwritten script identification problem is more complex than that of printed scripts due to the unequal variations among the authors, regions, topics, timing, etc. In contrast, the lack of datasets with several diverse images for various regional languages is a constraint for these challenges. Bengali is the second-most spoken language in India after Hindi and is most affected by the lack of dataset availability in the public domain for the development of a recognition system. The letters of the Bengali language are arranged in a family of complex scripts, either succeeding or following one another. Expanding the sequential training words through innovative techniques is necessary because acquiring data is expensive and challenging. To address this data shortage, a method for creating artificially handwritten Bengali sequence words is proposed in our research. For this, a sequential base generative adversarial network (SBGAN) model that draws inspiration from the generative adversarial network (GAN) is created to produce synthetic handwritten Bengali sequence words in order to address this data limitation. In this effort, the results of visual and quantitative examination show that the generated samples were realistic in nature and utilised to expand Bengali sequence words datasets.

**Keywords** GAN · Bengali script · Deep neural network

---

P. K. Samanta (✉) · A. Dutta · S. Biswas

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal 711103, India  
e-mail: [2020csp010.piush@students.iests.ac.in](mailto:2020csp010.piush@students.iests.ac.in)

A. Dutta  
e-mail: [arpita\\_dutta.rs2018@cs.iests.ac.in](mailto:arpita_dutta.rs2018@cs.iests.ac.in)

S. Biswas  
e-mail: [samit@cs.iests.ac.in](mailto:samit@cs.iests.ac.in)

## 1 Introduction

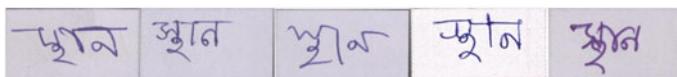
One of the essential human milestones is handwritten documentation. The majority of historical events were recorded in handwritten scripts. Historical handwritten data have several uses, particularly in education, health care, postal service, and finance, where data is still saved in handwritten form. Therefore, camera-captured handwritten script recognition is one of industry and academia's most exciting research areas. In this scenario, it is a need of the hour to design a well-organised system that can easily recognise handwritten scripts. Since India is a multi-lingual country, the handwritten script has several common shapes and features due to their shared origin. These common attributes across different scripts impose several challenges in the script recognition task. One of the Indian regional languages is Bengali, where the writers have various handwriting styles. The recent advancement of deep neural networks tackles those challenges more efficiently than standard feature-based approaches. However, those deep neural networks require large amounts of training instances to learn new features. These methods cannot quickly provide a viable solution with the limited number of handwritten documents. In reality, the number of publicly accessible datasets on Bengali handwritten documents is significantly very less. Bengali words are organised into a family of complex scripts where letters succeed or follow each other (shown in Fig. 1). The primary objective of the proposed approach is to generate these Bengali sequential words for expanding the variety of training data samples.

The overall contributions of our paper are as follows: we propose a deep neural network model, namely sequential base generative adversarial network (SGBAN), to generate handwritten images artificially. Our proposed approach achieves significant result while comparing with other state-of-the-art methods.

The rest of the paper is arranged as follows: the previous state-of-the-art methods for previous GAN approaches describes in Sect. 2. Section 3 represents a detailed description of our proposed method. Section 4 demonstrates the experimental results. In Sect. 5, we finally come to a conclusion.

## 2 Related Works

Data collection is a costly and time-consuming activity; therefore, available training data is usually limited. Here, we have chosen Bengali as the script to be recognised as part of handwritten script identification between different scripts. Additionally, the way the Bengali script operates is made more difficult by the fact that each writer's



**Fig. 1** Bengali word written by several authors. Due to their various shapes, certain word letters might be confused with others

handwriting is unique. Increasing the number of data samples with flexible handwritten text for training can help tackle this kind of difficulty. However, the number of publicly available datasets on Bengali handwritten documents is substantially lower. The artificial creation of handwritten Bengali sequential words is a possible option to compensate for the lack of a big dataset of the same, and this is the major topic of this research. Data augmentation is one option for growing training samples [19]. Generative models, which create pictures from any input, are another cutting-edge approach. This approach is still relatively new but is quickly gaining acceptance in the deep-learning community. Goodfellow et al. [9] initially presented the GAN architecture in 2014, where the generator creates artificial images that are as close as feasible to the genuine data. The discriminator's task is to separate these artificial images from the actual image data. In the end, competition between the two models leads to a situation in which the discriminator is unable to discern between bogus and actual data. Backpropagation can be used to train the aforementioned framework. One such instance of the usage of the Pix2pix framework with artificial modification is the work of Wang et al. [21]. The seldom availability of significant volumes of paired data in image-to-image translation jobs is a problem. The Pix2Pix model thus cannot be used in these circumstances. In order to address these issues, CycleGAN [24] offers a cycle-based uniform loss that makes it possible to map one-to-one between source to target domain even in the absence of paired instances. Alonso et al. [1] proposed a GAN architecture to create artificial images with strings of letters in them. Generator G and discriminator D make up the network. Additionally, convolutional layers and bidirectional LSTM layers are added at the end. Their effort was to create fixed-length and fixed-width handwritten strings in French and Arabic. When the created photos were combined with the already-existing dataset, the accuracy increased. A type of label conditioning was added to GANs in a recent study by Odena et al. [15]. In essence, the discriminator is coupled to a second fully connected network for class label prediction. Basu et al. [3] compare the U-Net and Pix2Pix frameworks for document picture binarization using a few competition datasets that are openly accessible. Other HOCR-related applications, such as line extraction [13] from text and adjacent component detachment [14], have also used GANs. An architecture suggested by Chang et al. [4] builds samples of one typeface from samples of another font. Compromises in the encoder network's architecture result in low-dimensional representations of input images. A transfer module creates the output font's feature representation. The HCCRGGoogleNet [23] classifier also evaluates the output images. Later comparisons with several additional GAN designs were demonstrated. Radford et al. proposed a non-conditional generative network, namely the Deep Convolutional Generative Adversarial Network (DCGAN) [17], which does not produce images utilising class knowledge in contrast to the AC-GAN. It is comparable to the original GAN created [2] in this sense. This study offers a topology for the GAN's architecture that may be used as a foundation for other efforts. InfoGAN [5] is a distinct non-conditional GAN that trains on a different mutual information loss function in order to enhance mutual knowledge from the GAN's variables to its inspections. DualGAN [22] is a dual-learning GAN that translates images from one domain to another.

In order to address this data shortage, a method for creating artificial handwritten Bengali compound characters is suggested in this study. In order to do this, a generative adversarial network (GAN)-based model is created, which draws influence from the recently published Auxiliary Classifier GAN (AC-GAN) model. For handwritten character identification, a unique dataset partitioning strategy is also devised to enhance the performance of the model proposed by Das et al. [6].

### 3 Proposed Method

The objective of our proposed SBGAN is to generate handwritten Bengali sequential word images. These artificial images can be utilised to train any deep neural network designed for handwritten script recognition. The discriminator  $D$  and the generator  $G$  are the two halves of our adversarial networks.  $G$  attempts to duplicate the original data images by synthesising fresh image samples. Next, the  $D$  tries to identify whether the produced image is genuine or false.  $G$  generates realistic artificial data samples  $G(v)$  by arbitrarily mapping a noise vector  $v$  to the training dataset. The generated artificial data samples are passed on to the  $D$  along with the corresponding real samples. It can be written as  $n \in L|n|$  denotes an image created from latent space,  $n$  denotes the number of dimensions, and  $G : G(v) \rightarrow L|n|$ , where  $v \in L|v|$  denotes latent vector noise. The task of discriminator  $D$  is to predict  $D : D(n) \rightarrow (0, 1)$  and assign a score to each image based on whether it is *genuine*(1) or *false*(0). A constructed loss function will reduce the information loss to compensate the discriminator for each incorrect prediction. Our method uses offline datasets to re-create the structures for handwritten data samples in Bengali, with their intricate complexities. The corresponding loss function  $LF_{D,G}$  for training the proposed model is defined in Eq. (1).

$$LF_{D,G} \leftarrow LF_{D,G} + \mathbb{E}_n [\log D(n)] + \mathbb{E}_v [\log(1 - D(G(v)))] \quad (1)$$

$\mathbb{E}_n$  is the expected value over all real samples, and  $\mathbb{E}_v$  is the expected value of overall generated artificial data samples  $G(v)$ .

#### 3.1 Pre-processing

The pre-processing stage for handwriting text generation frequently comprises image smoothing, skew and slant correction, image height, and pen stroke width normalisation to extract the word image from the handwritten document. The document image is initially converted into a binary image using the Otsu [16] binarisation technique. The binary image is smoothed using median filtering (shown in Fig. 2). The binary substance is divided into distinct text lines using a line segmentation technique. The border information and foreground seed components are then used to separate the

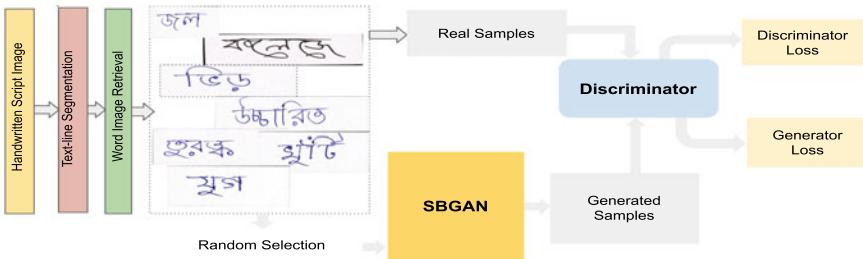
text lines. The Run Length Smoothing Algorithm (RLSA) [18], which isolates individual words as components, is applied to each text line after the text lines have been separated. Due to the recovered features' invariance under image height and pen stroke width, no modification will be made. The combination is built using form features extracted from the thinned images and a contour image that is always one pixel thick.

### 3.2 *Generation Network*

For our proposed network, handwritten Bengali script is regarded as a local process. The idea of adopting a completely convolutional generator architecture is closely followed by the suggested strategy, as employed in [7]. Instead of creating the entire sequence at once,  $G$  may be thought of as generating each Bengali alphabet separately. Hence, the effect of neighbouring characters will affect the entire architecture due to the overlapping nature of the receptive fields of the CNN [17]. As a result, the generator is perceived as being composed of several similar and class-conditioned generators. Each generator's class is represented by a single alphabetical letter from the Bengali language. Each generator generates a single patch with the needed character. Each layer utilised by upsampling is employed to increase the overlap between neighbouring letters. After being linked, a filter  $F$  is multiplied with a latent vector  $v$  of equal magnitude. Each character's  $F$  filter creates a region of the same size, and subsequent filters' receptive fields end up overlapping. The output letter may now be made in a variety of cursive sizes and styles similar to those shown here. It is necessary for distinct letters to come together to make linked sequences, and this is made possible by the overlap. Additionally, by learning relationships between nearby letters, the generator network is able to alter the appearance of the same letter based on its neighbours.

### 3.3 *Discrimination*

The function of  $D$  in a typical GAN is to reliably discriminate between samples of the original data and those that  $G$  has created.  $D$  is utilised to determine if an image is *genuine*(1) or *false*(0) in our proposed model. The discriminator has an architecture that is practically the exact opposite of that of  $G$ , and both are completely convolutional. The discriminator that analyses these images and creates output receives input from both real handwritten samples and generated samples. The weights of the generator and discriminator are then updated using this output in the loss function.



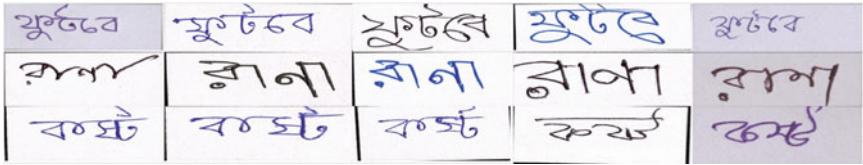
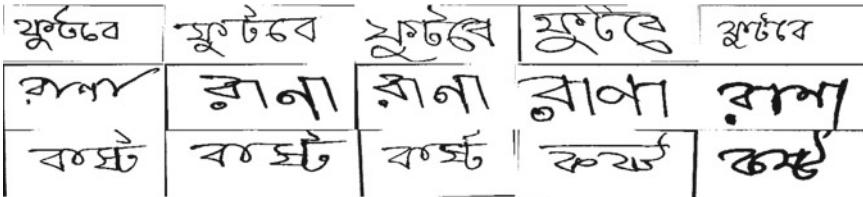
**Fig. 2** SBGAN architecture

### 3.4 SBGAN

Due to the diversity of the dataset, training on a basic GAN model is a sensitive and unstable process that may produce blurred outputs. Researchers have already experimented with various customizations and optimizations to generate various GAN versions, hence achieving improved learning stability. Various loss functions have been used to generate a wide range of unique image samples. Although the first is *DCGAN* that has different  $G$  and  $D$  architectures, which utilise a similar loss function as Standard GAN. Along with that, Wasserstein GAN (*WGAN*) [10] and Wasserstein GAN with Gradient Penalty are the other two implementations that were done (*WGAN – GP*). By utilising enhanced versions of the regular GAN, one may avoid concerns with balance and potential mode collapse that are frequently experienced during the training period of the classic SBGAN, which improves learning stability.

## 4 Experiments and Results

In this paper, we used a publicly accessible dataset of handwritten Bengali documents. Due to their shared ancestry, these scripts have comparable visual and structural elements. The creation of an annotated dataset is one of the most time-consuming and difficult activities. The pre-process step's execution results in the creation of the annotated dataset. We have utilised the publicly accessible benchmark dataset IIIT-INDIC-HW-WORDS [8] in accordance with the peculiarities of the artificial word generation. Word images are of varying horizontal and vertical resolutions. As a result, several images significantly differed from either mean vertical or horizontal resolution. Some images are excluded from the dataset during training. The results show that images with a  $128 \times 128$  performance are the best compared to other image sizes. The training procedure is significantly impacted by normalising these word images. In total, nearly 90k word images are present for the training, where 10k word images are identical. The models are trained over 800 epochs in our proposed SBGAN architecture. There are 128 batches in each cycle displaying some binarised

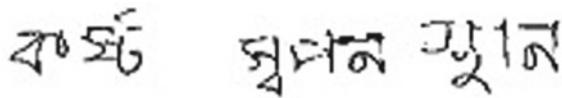
**Fig. 3** Real images**Fig. 4** Binarised images

samples for various training levels in Fig. 4. As our model is trained between 400 and 800 epochs, the image quality improves significantly (Figs. 5 and 6). Publicly accessible generated images are provided for each of the 10k image terms. The model's key strength is the ability to create effective results for both straightforward and intricately designed figures. Other models get decent results for the basic classes but consistently subpar outcomes for the more complicated shaped classes.

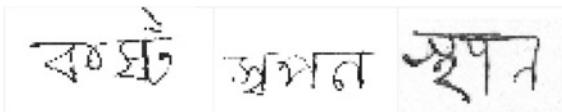
The Fréchet Inception Distance (FID) score [11] is usually used to evaluate the achievement of our proposed model. This is the most common statistic for evaluating the effectiveness of GAN models. Both the quality and the variation of the images produced are considered. The Inception V3 model is used to compute FID score between genuine and artificially created images [20]. In essence, it evaluates the similarities between the two image dataset. The lesser FID score signifies the similarity between the produced and real images (Fig. 3).

FID score is formed between two multidimensional Gaussian distributions. This is how FID differs from the formerly popular Inception score measure. Only artificial images are used to produce the Inception score, which uses parameters like dataset variance. Different characteristics of the genuine and artificial images are represented by their FID score. In the form of feature matrices, these distributions would be produced by a CNN model. These are produced using the Inception V3 model's intermediate layers, which were developed using ImageNet data. Additionally, using the intermediate layers makes it possible to get scores nearly matching human judgement. For each experiment, the FID score was computed using the entire dataset in comparison with an equivalent number of produced samples, or around 90k samples. In order to measure mode collapse, Geometric Score (GS [12]) analyses the geometrical characteristics of the basic actual and false data manifolds. GS was computed on 10k actual vs 10k generated samples with trivial parameters. FID was computed in experiments using various hyperparameters per ten iterations. The best

**Fig. 5** Generated images using SGBAN after 400 epochs



**Fig. 6** Generated images using SGBAN after 800 epochs



**Table 1** Performance evaluation for GANs variants using benchmark dataset IIIT-INDIC-HW-WORDS [8]

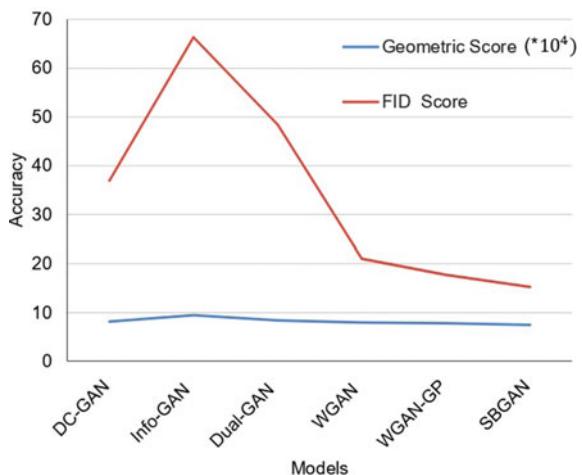
Model	FID score	Geometric score
DCGAN	36.92	$8.14 \times 10^4$
InfoGAN	66.38	$9.46 \times 10^4$
DualGAN	48.42	$8.41 \times 10^4$
WGAN	21.02	$7.96 \times 10^4$
WGAN-GP	17.74	$7.82 \times 10^4$
SGBAN	15.23	$7.46 \times 10^4$

FID was selected from all trials run, and GS was also calculated for this model set-up. The verification of textual material was based on visual inspection. FID scores correlate with human perception of the visual quality of produced artificial image samples, and GS scores also support these findings.

We also compared the FID for a few different GAN models, including DCGAN, InfoGAN, and DualGAN. The dataset restructuring methods have been used to train these models in order to provide a fair comparison. InfoGAN is not a conditional GAN, in contrast to the other GANs tested on the dataset. Image-to-image conversion as generation is carried out by DualGAN using a number of generators and discriminators. Therefore, compared to the architecture described in this article, each of these GANs has quite different designs. The WGAN-GP architecture recorded the lowest FID score, which was 15.74 at its lowest. This rating is on par with the most recent scores for other languages. This is also consistent with the qualitative study, which demonstrated that the other two designs were superior to DCGAN in terms of picking up and creating more detailed samples. The lowest FID score that could have been obtained by DCGAN was 21.55, but it was 17.98 with WGAN. The WGAN-GP model had the lowest score, indicating that it was the best model out of the three, and GS values were also correlated with FID scores. In order to compare the performance of the various GAN implementations, including our proposed approach, Table 1 displays FID Scores and Geometric Scores, and Fig. 7 refers corresponding plotting to the basics of the FID and Geometric scores.

The suggested strategy is unable to fully develop the ability to discern between the minute features in such a sequence, leading to the production of compressed

**Fig. 7** Performance plotting with GANs Variants



forms of these sequential words that do not adequately capture all the information. Thus, it can be concluded that an output width of 128 is insufficient to accommodate all the characters needed to be created for sequential words with more alphabets. However, generators with varying length outputs should be used to produce entire Bengali words. Given the facts, we may draw the conclusion that Bengali letters are more diverse than English letters, where there is a solid association between the widths of a letter's lowercase and uppercase variants. In the vocabulary, they are also regarded as distinct letters. On the other hand, Bengali letters do not adhere to any such pattern; instead, the architecture must become used with the fact that a single letter might fluctuate in breadth and shape based on its surrounding characters. In the case of the Bengali language, this also presents a barrier to implementing a generator with a changeable output length.

## 5 Conclusion

In this paper, we created an artificial dataset of handwritten Bengali words. The development of the Bengali sequential word requires the synthesis of linked components, which is made possible by the convolutional nature of structures. A single generator may produce various sequences, something that has never been done previously only for Bengali, and assessments were done that can be used for future research. The proposed method produced better results and yielded FID scores equivalent to other state-of-the-art models while considering the complexity of Bengali handwriting samples. As opposed to conventional GANs, the proposed SBGAN will break sequences into basic letters before being supplied.

## References

- Alonso E, Moisset B, Messina R (2019) Adversarial generation of handwritten text images conditioned on sequences. In: 2019 International conference on document analysis and recognition (ICDAR), pp 481–486. IEEE
- Ba H (2019) Improving detection of credit card fraudulent transactions using generative adversarial networks. arXiv preprint [arXiv:1907.03355](https://arxiv.org/abs/1907.03355)
- Basu A, Mondal R, Bhowmik S, Sarkar R (2020) U-net versus pix2pix: a comparative study on degraded document image binarization. *J Electron Imaging* 29(6):063019–063019
- Chang B, Zhang Q, Pan S, Meng L (2018) Generating handwritten chinese characters using cyclegan. In: 2018 IEEE winter conference on applications of computer vision (WACV), pp 199–207. IEEE
- Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: interpretable representation learning by information maximizing generative adversarial nets. *Adv Neural Inf Process Syst* 29:1–9
- Das A, Choudhuri A, Basu A, Sarkar R (2022) Generation of a synthetic handwritten bangla compound character dataset using a modified conditional gan architecture. *Multimedia Tools Appl* 1–23
- Fogel S, Averbuch-Elor H, Cohen S, Mazor S, Litman R (2020) Scrabblegan: semi-supervised varying length handwritten text generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4324–4333
- Gongidi S, Jawahar C (2021) iiit-indic-hw-words: a dataset for indic handwritten text recognition. In: Document analysis and recognition—ICDAR 2021: 16th international conference, Lausanne, Switzerland, Sept 5–10, 2021, Proceedings, Part IV 16, pp 444–459. Springer
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144
- Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC improved training of wasserstein gans. *Adv Neural Inf Process Syst* 30:1–11
- Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30:1–12
- Khrulkov V, Oseledets I (2018) Geometry score: a method for comparing generative adversarial networks. In: International conference on machine learning, pp 2621–2629. PMLR
- Kundu S, Paul S, Bera SK, Abraham A, Sarkar R (2020) Text-line extraction from handwritten document images using gan. *Expert Systems with Applications* 140:112916
- Mondal R, Bhowmik S, Sarkar R (2020) Tseggan: a generative adversarial network for segmenting touching non-text components from text ones in handwriting. *IEEE Trans Instrum Meas* 70:1–10
- Odena A, Olah C, Shlens J (2017) Conditional image synthesis with auxiliary classifier gans. In: International conference on machine learning, pp 2642–2651. PMLR
- Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9(1):62–66
- Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
- Roy PP, Pal U, Lladós J (2008) Morphology based handwritten line segmentation using foreground and background information. In: International conference on frontiers in handwriting recognition, pp 241–246
- Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. *J Big Data* 6(1):1–48
- Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826

21. Wang TC, Liu MY, Zhu JY, Tao A, Kautz J, Catanzaro B (2018) High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8798–8807
22. Yi Z, Zhang H, Tan P, Gong M (2017) Dualgan: unsupervised dual learning for image-to-image translation. In: Proceedings of the IEEE international conference on computer vision, pp 2849–2857
23. Zhong Z, Jin L, Xie Z (2015) High performance offline handwritten chinese character recognition using googlenet and directional feature maps. In: 2015 13th international conference on document analysis and recognition (ICDAR), pp 846–850. IEEE
24. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232

# A Transfer Learning Approach to Indian Currency Coin Recognition



R. Ahila Priyadharshini, M. Vijayraj, and R. Dhanushraj

**Abstract** Automatic coin recognition and identification system plays vital role in vending, slot machine and in several banking related equipments. Most people require the ability to recognize the value of various forms of currency in order to function in daily life. In order to automate financial transactions, computers must be capable of performing such recognition. The majority of current Indian currency coin denominations are of comparable size, making the identification process more difficult. In this work, we applied a transfer learning approach for recognizing Indian currency coins. The pretrained model used is VGG19. The model is tested on two datasets: our own dataset named “Mepco coin dataset” and Indian Coin Denomination dataset (ICDD) and report an accuracy of 92.8% and 95%.

**Keywords** Coin recognition · Deep learning · VGG19

## 1 Introduction

Machine vision systems must be able to perform and identify objects in any position, size, or appearance [1] in the same way that visual systems can effectively differentiate between objects. Object recognition refers to the identification of an object within a digital image based on various discriminative features. In this study, object recognition techniques were applied to coin recognition. Automatic coin value recognition has a variety of uses [2]. For example, slot machines for gambling and vending machines for snacks, beverages, banks, tickets, supermarkets, etc. To reduce human error and expedite the repetitive process of counting and sorting a large number of coins, automated coin systems are utilized. Measurement of the coin’s physical characteristics, such as weight, thickness, diameter, and magnetism, is the primary strategy of modern coin recognition systems. However, counterfeit coins with identical physical characteristics are acknowledged in these systems and are commonly

---

R. A. Priyadharshini (✉) · M. Vijayraj · R. Dhanushraj

Department of ECE, Mepco Schlenk Engineering College, Sivakasi 626005, India  
e-mail: [rahila@mepcoeng.ac.in](mailto:rahila@mepcoeng.ac.in)

encountered. To improve the recognition rate's accuracy, the coin recognition system increasingly takes the coin's visual characteristics into account. In the recognition system, the coin's image is acquired, analyzed, and processed.

The models available in the literature for currency coin recognition fall on machine learning and deep learning-based models. In machine learning schemes, features are manually calculated before a learning mechanism or decision-making is constructed. The machine learning approaches rely on the manually calculated characteristics of the original currency coin images. In template matching approaches, the manually-crafted features of the query image are matched with the manually-crafted features derived from the template currency coin images in the database. Chetan and Vijay [3] used template matching to identify different denominations of Indian coins. This involves two phases of matching an input coin image to the images in database. Finding coin matches in the radius database is the first step of the process. In the second phase of the template matching process, the edges of the input coin images and those from the matching radius database are correlated. Shen et al. [4] represented texture information for coin recognition using Gabor wavelets and a local binary pattern (LBP) feature. Using a concentric ring arrangement, the coin image is divided into a number of small segments. Then the statistics of Gabor filtered coefficients and LBP values within each segment are concatenated to yield an image feature vector. It is proposed that a circular shift operator will make Gabor features rotation invariance. Using the nearest neighbor classifier, a test coin is classified. Lin and Jhuo [5] proposed a coin recognition system that extracts texture features from the coin image and create dictionaries using bag of words (BoW) approach. Xu et al. [6] used SIFT algorithm for coin recognition. The SIFT algorithm is capable of handling image rotations, scaling, and illumination. Consequently, it can distinguish coins of comparable size. Roomi and Rajee [7] applied Fourier transform on the coin image to diminish variations on the coin's surface, such as the effect of light reflection. Then the transformed features are classified by multi-layered BP neural network. Kim et al. [8] proposed spatial structure-based coin recognition. To evaluate the coin's structure, the rotation and flipping robust region binary patterns (RFR) are utilized. The magnitudes of the gradients are computed in a coin image and RFR are then extracted to improve coin recognition accuracy.

Recently deep learning techniques have gained attention on object recognition, computer vision applications such as plant disease classification [9], medicinal plant identification [10], and biometric recognition [11]. Capece et al. [12] used pretrained Alexnet to classify Euro coins. Kim and Pavlovic [13] proposed a Convolutional Neural Network (CNN)-based method for the automatic recognition of Imperial Roman coins. The research employs CNN models in a hierarchical framework for coin classification tasks. Tajane et al. [14] utilized a pretrained CNN, namely AlexNet, for the recognition of Indian currency coins. The model is trained on more than 1600 images and is capable of classifying images into four object categories, such as coins of one, two, five, and ten rupees. Xiang and Yan [15] identified fast-moving coins in digital videos using deep learning techniques, particularly a blend of long short-term memory (LSTM) and CNN. From the literature, it is clear that limited research works are available for recognizing Indian coin denominations. Only one

dataset is publicly available with total of 900 coins for different denomination such as One (₹1), Two (₹2), Five (₹5), Ten (₹10), and Twenty (₹20) rupee coins, so, we intend to introduce Mepco coin dataset with 3398 coins with the above mentioned denominations and attempt to apply a transfer learning approach to Indian Coin Denomination Dataset (ICDD) dataset having five denominations. Moreover, the approach is applied on our own coin dataset, Mepco coin dataset.

## 2 Materials and Methods

This section discusses about the datasets used and the proposed transfer learning approach used in this research work.

### 2.1 *Indian Coin Denomination Dataset (ICDD)*

The dataset includes images of the Indian currency coin denominations of One (₹1), Two (₹2), Five (₹5), Ten (₹10), and Twenty (₹20) rupee coins, as well as their variations [16]. The dataset only includes images of the denomination-bearing tail side of each coin. This dataset contains a total of 30 different coins, including 7 numbers of one rupee coins, 4 numbers of two rupees coins, 9 numbers of five rupees coins, 6 numbers of coins worth ten rupees, and 4 numbers of coins worth twenty rupees each coin was photographed from five distinct viewing angles and under three distinct lighting conditions, with each combination of viewing angle and lighting condition being photographed twice. The dataset includes a total of 900 images, including 210 images from the rupee one class, 120 images from the rupee two class, 270 images from the rupee five class, 180 images from the rupee ten class, and 120 images from the rupee twenty class. Each image in the dataset is of the size  $3000 \times 4000$  pixels. The sample images of ICDD dataset is shown in Fig. 1.

### 2.2 *Mepco Coin Dataset*

This dataset contains a total of 3398 Indian currency coin images with different denominations such as One (₹1), Two (₹2), Five (₹5), Ten (₹10), and Twenty (₹20) rupee coins. The dataset includes images of the denomination-bearing both head and tail side of each coin. The coin images are captured under different lighting conditions (very low light and high power light), natural light, different viewing distances, different coin positions, different coins angles, and different backgrounds. All coins images are taken with the vertical resolution and horizontal resolution of 72 ppi. The image count in each denomination is shown in Table 1. The sample images of Mepco coin dataset is shown in Fig. 2.



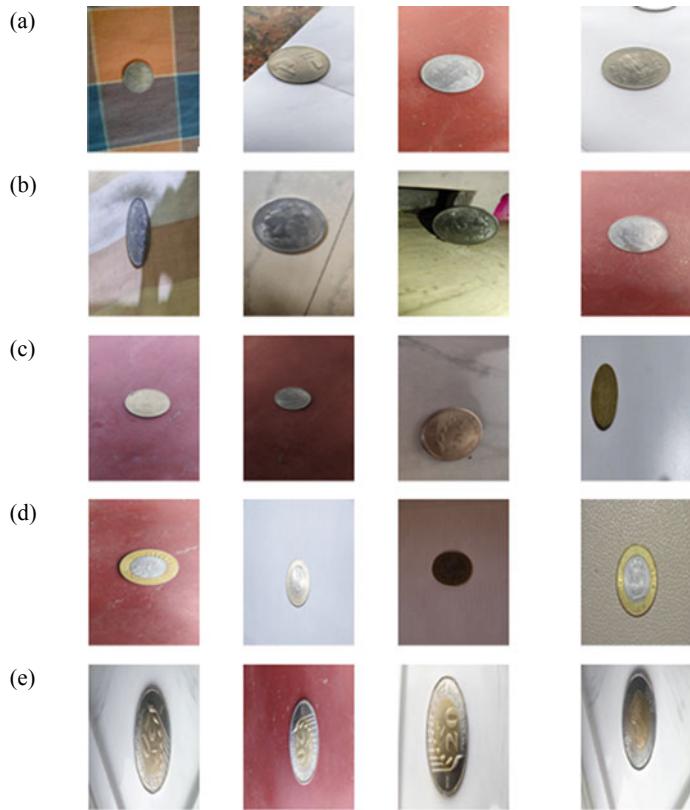
**Fig. 1** Different denominations of Indian currency coins in ICDD dataset. **a** Rupee one (₹1), **b** Rupees two (₹2), **c** Rupees five (₹5), **d** Rupees ten (₹10), **e** Rupees twenty (₹20)

**Table 1** Details of Mepco coin dataset

Coins	# Images
Rupee one (₹1)	970
Rupees two (₹2)	1087
Rupees five (₹5)	895
Rupees ten (₹10)	307
Rupees twenty (₹20)	139

### 2.3 Transfer Learning Approach

Transfer learning is the application of an existing model to the solution of distinct but related problems. Transfer learning reduces the need for a massive dataset, reduces training time, and improves performance in the majority of cases. “VGG19 is a 19 layer deep convolutional neural network with 16 convolution layers, and 3 fully connected layers and is designed to classify images into 1,000 object categories. VGG19 is trained on the ImageNet database, which contains one million images divided into one thousand categories. Due to the use of multiple  $3 \times 3$  filters in each convolutional layer, it is a very popular method for image classification. The first sixteen convolutional layers are used for feature extraction, while the next three fully connected layers are responsible for classification. There are 5 groups of layers used for feature extraction, with a max-pooling layer succeeding each group. This model receives an input image of size  $224 \times 224$  and outputs the label of the object in the image”. To train with our coin dataset and categorize the Indian currency coin



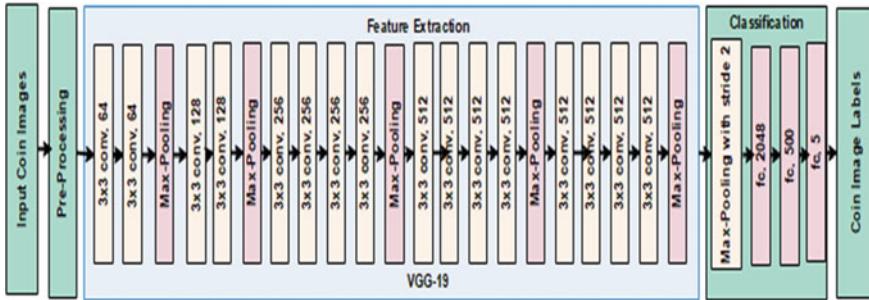
**Fig. 2** Different Denominations of Indian Currency coins in Mepco coin dataset. **a** Rupee one (₹1), **b** Rupees two (₹2), **c** Rupees five (₹5), **d** Rupees ten (₹10), **e** Rupees twenty (₹20)

images, we use VGG19 as our base model and freeze the weights in all the layers except the fully connected layers and retrain the model using the coin dataset. The flow diagram for transfer learning approach is shown in Fig. 3.

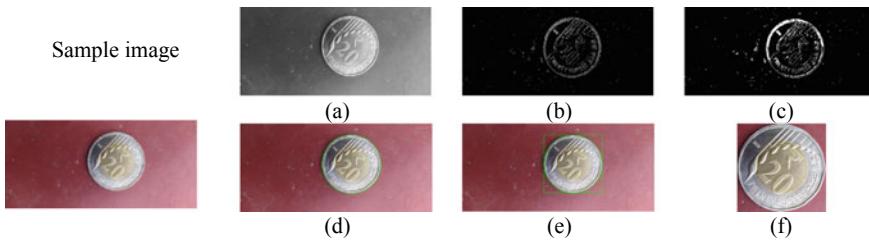
### 3 Results and Discussion

To obtain the region of interest (coin) from an image, the coin's contour is traced. First, grayscale images are converted to binary using the Ostu thresholding algorithm. Following the closing operation on the binary thresed image, the ROI (coin) is cropped from the image and the contour is traced. Figure 4 depicts the steps involved in the preprocessing of an example coin image.

The cropped coin images are resized to  $224 \times 224$  in order to accommodate the larger input size of the VGG19 pretrained model. We freeze the weights of all layers



**Fig. 3** Flow diagram for transfer learning approach



**Fig. 4** Preprocessing in a sample image. **a** Grayscale conversion, **b** binary thresholding, **c** morphological closing, **d** contour detection, **e** bounding box (ROI), **f** cropping

except those with complete connections. The number of neurons in the output layer is maintained at 5, and the RMSprop optimizer is used to conduct experiments. The initial learning rate is set as 0.001 and  $\epsilon$  as  $1 \times 10^{-7}$ . The experiment is conducted over twenty epochs. Tables 2 and 3 display the number of images used for training and testing and the performance metrics for the Mepco coin dataset and the ICDD dataset, respectively.

Figure 5 depicts the confusion matrix for the two datasets mentioned above. The proportion of relevant coin images among the retrieved coins is known as precision, whereas the proportion of relevant coin images that were retrieved is known as recall. The F1-score indicates the harmonic mean of both measures. Due to their structural similarity, in the Mepco coin dataset, one rupee coins are confused with two and five rupee coins, and similarly, two rupee coins are confused with one and five rupee coins. One rupee coin is confused with five and twenty rupee coins in the ICDD dataset, while two rupee coins are confused with one rupee coins. The obtained overall accuracy for the Mepco coin and ICDD datasets is 92.8% and 95.0%, respectively. Table 4 compares the performance of the ICDD dataset to that of the existing method. Our transfer learning approach is better than the existing approach.

**Table 2** Performances measures of Mepco coin dataset

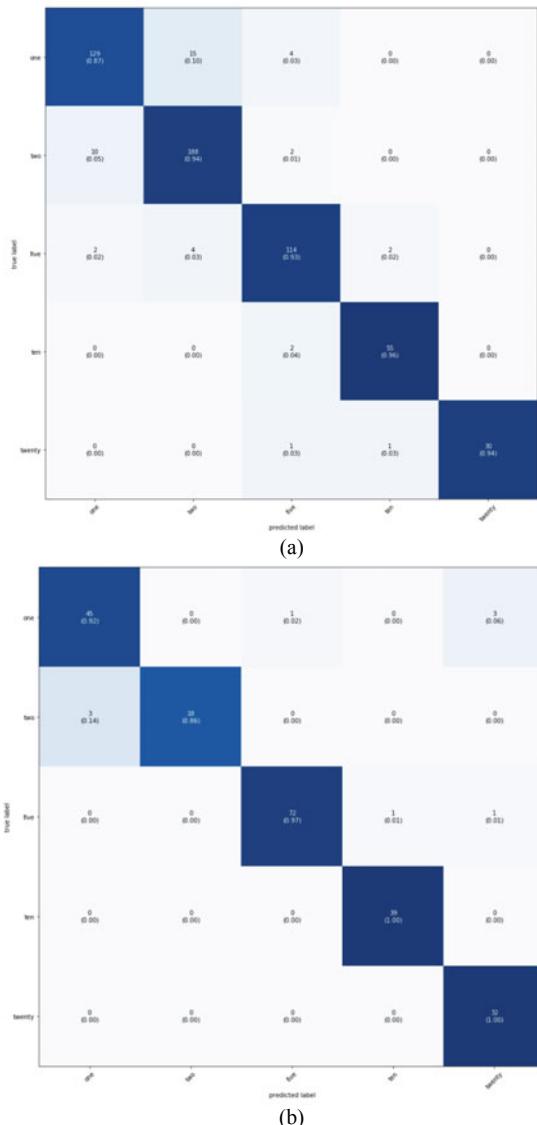
Coins	# Training images	# Testing images	# Misclassification	Precision	Recall	F1-score
Rupee one (₹1)	822	148	19	0.91	0.87	0.89
Rupees two (₹2)	887	200	12	0.91	0.94	0.92
Rupees five (₹5)	773	122	8	0.93	0.93	0.93
Rupees ten (₹10)	250	57	2	0.95	0.96	0.96
Rupees twenty (₹20)	107	32	2	1.00	0.94	0.97

**Table 3** Performances measures of ICDD dataset

Coins	# Training images	# Testing images	# Misclassification	Precision	Recall	F1-score
Rupee one (₹1)	161	49	4	0.94	0.92	0.93
Rupees two (₹2)	99	21	3	1.00	0.86	0.92
Rupees five (₹5)	196	74	2	0.99	0.97	0.98
Rupees ten (₹10)	141	39	0	0.97	1.00	0.99
Rupees twenty (₹20)	88	32	0	0.89	1.00	0.94

**Fig. 5** Confusion matrices.

**a** Mepco coin dataset,  
**b** ICDD dataset

**Table 4** Performance comparison of ICDD dataset with existing method

Method	Accuracy (%)
CNN [17]	90.55
Our method	95

## 4 Conclusion

In this work, a new database containing 3398 Indian currency coin images are created and that can be available for training a machine learning model to classify Indian currency coins by denomination. We studied the efficiency of transfer learning approach for coin recognition on two datasets namely Mepco coin dataset and ICDD dataset and report an accuracy of 92.8% and 95%, respectively. This work's scope can be expanded by deploying this application in any microcontroller interfaced with a camera and a display, and it can also be deployed in real-time embedded systems.

## References

1. Priyadarshini RA, Arivazhagan S (2013) Object recognition based on local steering kernel and SVM. *Int J Eng* 26(11):1281–1288
2. Lin W-J, Jhuo S-S (2016) Coin recognition based on texture classification on ring and fan areas of the coin image. In: IEEE instrumentation and measurement technology conference proceedings, vol 12, Jul 2016, pp 25–30
3. Chetan BV, Vijaya PA (2013) A robust method of image based coin recognition. In: Kumar MARS, Kumar T (eds) Proceedings of international conference on advances in computing, advances in intelligent systems and computing, vol 174. Springer, New Delhi. [https://doi.org/10.1007/978-81-322-0740-5\\_110](https://doi.org/10.1007/978-81-322-0740-5_110)
4. Shen L, Jia S, Ji Z, Chen WS (2011) Extracting local texture features for image-based coin recognition. *IET Image Proc* 5:394–401
5. Lin W-J, Jhuo S-S (2016) Coin recognition based on texture classification on ring and fan areas of the coin image. In: 2016 IEEE international instrumentation and measurement technology conference proceedings, pp 1–6. <https://doi.org/10.1109/I2MTC.2016.7520507>
6. J. Xu, G. Yang, Y. Liu and J. Zhong, "Coin Recognition Method Based on SIFT Algorithm," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), 2017, pp. 229–233, doi: <https://doi.org/10.1109/ICISCE.2017.57>.
7. Roomi SMM, Rajee RBJ (2015) Coin detection and recognition using neural networks. In: 2015 International conference on circuits, power and computing technologies (ICCPCT-2015), pp 1–6. <https://doi.org/10.1109/ICCPCT.2015.7159434>
8. Kim S, Lee SH, Ro YM (2015) Image-based coin recognition using rotation-invariant region binary patterns based on gradient magnitudes. *J Vis Commun Image Represent* 32:217–223. <https://doi.org/10.1016/j.jvcir.2015>
9. Ahila Priyadarshini R, Arivazhagan S, Arun M et al (2019) Maize leaf disease classification using deep convolutional neural networks. *Neural Comput App* 31:8887–8895. <https://doi.org/10.1007/s00521-019-04228-3>
10. Ahila Priyadarshini R, Arivazhagan S, Arun M (2021) Ayurvedic medicinal plants identification: a comparative study on feature extraction methods. In: Singh SK, Roy P, Raman B, Nagabhushan P (eds) Computer vision and image processing. In: CVIP 2020. Communications in computer and information science, vol 1377. Springer, Singapore. [https://doi.org/10.1007/978-981-16-1092-9\\_23](https://doi.org/10.1007/978-981-16-1092-9_23)
11. Ahila Priyadarshini R, Arivazhagan S, Arun M (2021) A deep learning approach for person identification using ear biometrics. *Appl Intell* 51:2161–2172. <https://doi.org/10.1007/s10489-020-01995-8>

12. Capece N, Erra U, Ciliberto AV (2016) Implementation of a coin recognition system for mobile devices with deep learning. In: 2016 12th international conference on signal-image technology & internet-based systems (SITIS), pp 186–192. <https://doi.org/10.1109/SITIS.2016.37>
13. Kim J, Pavlovic V (2015) Discovering characteristic landmarks on ancient coins using convolutional networks. CoRR, abs/1506.09174
14. Tajane AU, Patil JM, Shahane AS, Dhulekar PA, Gandhe ST, Phade GM (2018) Deep learning based Indian currency coin recognition. In: International conference on advances in communication and computing technology (ICACCT), pp 130–134. <https://doi.org/10.1109/ICACCT.2018.8529467>
15. Xiang Y, Yan WQ (2021) Fast-moving coin recognition using deep learning. *Multimed Tools Appl* 80:24111–24120. <https://doi.org/10.1007/s11042-021-10857-5>
16. Chauhan Y, Singh P (2021) Indian coin denomination dataset (ICDD). IEEE DataPort. <https://doi.org/10.21227/t63t-5j24>
17. Chauhan Y, Singh P (2021) A novel convolutional neural network for classifying Indian coins by denomination. <https://doi.org/10.31224/osf.io/znxrg>

# SDinIWTrack: A Novel Database for Training Self-driving Vehicles



Swati Chowdhuri, Sriparna Banerjee, and Supriya Mondal

**Abstract** Object tracking is an emerging research topic in the computer vision field and overwhelming research progress has been achieved in this field in recent years because of its real-life significance. Designing efficient self-driving vehicles is one of the most popular applications in this research domain. Although commendable research advancements have been achieved in this research field but still there are various unexplored areas in which further research can be conducted. The performance efficiencies of the automated tracking methods are largely dependant on the databases which are used to train them apart from the methods' architectures. Owing due to the importance of databases in this research domain, here we have conducted a comprehensive survey on existing object tracking databases. The main contribution of this paper lies in the creation of a novel database namely, SDinIWTrack for training self-driving vehicles to drive safely during challenging conditions and to avoid undesirable fatal accidents. This database includes thousands of annotated sequences of road scenes captured during inclement weather conditions for, e.g., sunny, rainy, foggy, etc. and under varied illumination conditions for, e.g., uniform illumination (during daytime) and spatially variant illumination (during night). To the best of our knowledge, SDinIWTrack is the first database which is created with this objective. The sequences of road scenes included in this database are collected from various publicly available sources and are annotated by a group of annotators.

**Keywords** Self-driving vehicles · Object detection and tracking · Database · SDinIWTrack · Survey · Inclement weather · Varied illumination

---

S. Chowdhuri (✉) · S. Mondal

EEE Department, Institute of Engineering and Management, Kolkata 700091, India  
e-mail: [swati.chowdhuri.iemcal@gmail.com](mailto:swati.chowdhuri.iemcal@gmail.com)

S. Banerjee

ETCE Department, Jadavpur University, Kolkata 700032, India

## 1 Introduction

Object detection and tracking is one of the most popular area of research in the computer vision field as it has found its use in various important applications, one of the most significant of which is self-driving vehicles.

Due to the immense popularity of this research area in real world as well as due to the overwhelming performances achieved by deep neural networks in computer vision tasks in recent years because of increase in data availability and enhancement of computational capabilities of systems, many researchers have designed several automated object tracking methods.

The performance efficiencies of these automated object tracking methods are primarily depend on two factors namely, a. The method's architecture and b. Database on which these models are trained.

The characteristics of the images included in the databases largely determine the performances of the automated methods as these methods classify, detect, and track target objects during validation as well as test phases based on the knowledge gained by the models from the characteristics of the images given as inputs during training phase.

Owing due to the significance of the databases in any computer vision tasks, in this paper we have conducted a comprehensive survey on databases existing in this research area.

Apart from the survey, the main contribution of this work lies in the creation of novel database namely, SDinIWTrack which includes thousands of annotated sequences of road scenes captured while driving through different terrains during several weather conditions like sunny, foggy, rainy, etc. as well as under varied illumination conditions with the objective to perform efficient training of self-driving vehicles to drive satisfactorily during inclement weather conditions and under varied illumination conditions to avoid undesired fatal accidents.

Details of comprehensive survey which is conducted on existing databases are given briefly in Sect. 2 while detailed description of the novel SDinIWTrack database is given in Sect. 3. This work is concluded in Sect. 4 where the future scope of work is also highlighted.

## 2 Brief Details of Existing Object Tracking Databases

In this section, we have performed comprehensive survey on existing object tracking databases and highlighted relevant details like their contents, applications, etc. After giving the details of all these databases, we have also performed comparative analysis among those databases and represented the analysis in tabular form in Table 4

**Table 1** Year-wise development achieved in PASCAL VOC datasets

Year	Object categories	Description
2005	Bicycles, Cars, Motorbikes, and People	Frames included in this database are taken from several publicly available sources and contain no challenging cases. Hence, VOC 2005 is no longer used for bench-marking tracking methods
2006	Bicycles, Bus, Car, Cat, Cow, Dog, Horse, Motorbike, Person, and Sheep	Performing localization of objects present in Microsoft Research Cambridge (MSRC) dataset [2] is often easier to perform compared to images taken from Flickr [3] as those images are most concentrated on target objects
2007	Person, Bird, Cat, Cow, Dog, Horse, Sheep, Airplane, Bicycle, Boat, Bus, Car, Motorbike, Train, Bottle, Chair, Dining table, Potted plant, Sofa, and tv/monitor	The number of object categories of VOC datasets is fixed to 20 from this year onwards The final annotations for testing data is released in this year
2008	Same as PASCAL VOC 2007	Test data annotations are made publicly available
2009	Same as PASCAL VOC 2007	Test data annotations are made publicly available No difficult flags were given for additional images
2010	Same as PASCAL VOC 2007	The method of Average Precision computation is changed in PASCAL VOC 2010 compared to the earlier years. Test data annotations are made publicly available
2011	Same as PASCAL VOC 2007	Layout annotations are introduced for some images belonging to object category, ‘people’
2012	Same as PASCAL VOC 2007	No new layout annotation is performed

## 2.1 PASCAL VOC Datasets (2005–2012) [1]

The main aim of PASCAL VOC project is to design datasets comprising of frames containing annotated objects belonging to various object categories for performing several object detection tasks. Year-wise development achieved in the databases developed under this project is given in tabular form in Table 1.

## 2.2 ImageNet: A Large-Scale Hierarchical Image Databases (ILSVRC) (2010–2017) [4]

Several databases were designed under ILSVRC challenge over the period of 2012–2017. These databases were created to benchmark the object recognition, classification, and detection capabilities of designed computer vision algorithms irrespective of the inter-variations and intra-variations in characteristics of objects belonging to different object categories. Advancements introduced in these databases each year are given in Table 2.

**Table 2** Year-wise advancements introduced in ILSVRC datasets

Year	Description of databases	Objective
2010	Training set used in this challenge is composed of images belonging to 1000 object categories included in ImageNet database [5]. 200,000 images containing either one or none of the objects belonging to those 1000 object categories are included in the validation and test sets. The images belonging to the validation and test sets are taken from Flickr [3]	To assign an overall error score to each of the algorithms participating in this challenge based on how accurately they can identify the object categories. The error is calculated by computing the differences between predicted object categories and actual object categories and overall error score is evaluated by averaging the errors obtained for all test images. The evaluation is performed in two ways namely, a. Hierarchical evaluation (when evaluation is performed considering the hierarchical structure of object categories) and non-hierarchical evaluation (where evaluation is performed considering all object categories as equal)
2011	Training set used in this challenge is composed of images belonging to 1000 object categories included in ImageNet database [5]. 150,000 images containing either one or none of the objects belonging to those 1000 object categories are included in the validation and test sets. The images belonging to the validation and test sets are taken from Flickr [3]	In addition to classification capabilities, the algorithms participating in this challenge are also evaluated according to their object localization capabilities. The classification capabilities of participating algorithms are done similarly as done in ILSVRC 2010 challenge and error for object localization task is measured by calculating how accurately the predicted bounding box co-ordinates of detected objects overlap with the actual bounding box co-ordinates of detected objects. If the overlap percentage is less than 50%, then the error is considered to be 1
2012	Training set used in this challenge is composed of images belonging to 1000 object categories included in ImageNet database [5]. 150,000 images containing either one or none of the objects belonging to those 1000 object categories are included in the validation and test sets. The images belonging to the validation and test sets are taken from Flickr [3]	In addition to classification and localization tasks, the algorithms participating in this challenge are also evaluated based on their capabilities to perform fine-grained classification using images of dogs belonging to 100+ dog categories. This evaluation is performed based on how accurately those algorithms predict the categories of dogs detected with bounding boxes in test images

(continued)

**Table 2** (continued)

Year	Description of databases	Objective
2013	<p>In ILSVRC 2013 challenge, two datasets were used for performing two different tasks. For performing detection, a database is designed comprising of fully annotated and labeled images belonging to 200 basic level categories which are chosen based on varied factors like different levels of image clutter, object scale, etc. This dataset is created mostly focusing on people detection and comprises of 12,125 training images (out of which 9788 images contain people), 20,121 images for validation and 40,152 testing images</p> <p>The dataset used for performing classification and localization tasks in ILSVRC 2013 challenge is similar to the dataset used in ILSVRC 2012 challenge</p>	<p>The algorithms participating in this challenge are evaluated similarly for performing classification and localization tasks as done in ILSVRC challenges conducted on previous years but for detection task, the algorithms are evaluated based on the set of annotations (where <math>c_i</math> represents class labels and <math>b_i</math> and <math>s_i</math> signify bounding box co-ordinates and confidence scores). The algorithms were penalized when annotations for an object is not generated or duplicate detection were performed</p>
2014	<p>In this year's challenge, datasets used for performing detection and classification and localization tasks are similar to the databases used in ILSVRC 2013 challenge but the training set used for detection task is extended by including 60,658 images which are obtained from Flickr [3]</p>	<p>The algorithms participated in this challenge are evaluated similarly as done in ILSVRC 2013 challenge for both detection as well as classification and localization tasks</p>
2015	<p>Dataset used in this challenge for performing detection task is similar to the dataset used in ILSVRC 2014 challenge but the test set is partially refreshed using new images while for performing object localization task, dataset used is similar to ILSVRC 2012 challenge. In addition to these datasets, two more datasets are used in ILSVRC 2015 challenge one for performing detection of objects from video frames. The videos included in this database contain objects belonging to 30 object categories which is a subset belonging to 200 basic object categories considered for object detection task. Another existing dataset namely, Places2 dataset [6] is used for performing scene classification task</p>	<p>The algorithms are evaluated similarly for object detection, classification, and localization tasks as done in previous challenges</p> <p>For object detection from video task, the algorithms are evaluated similarly as done in object detection task but using a set of annotations (<math>f_i, c_i, b_i, s_i</math>) (where <math>f_i, c_i</math> represent frame number and class labels. <math>b_i</math> and <math>s_i</math> signify bounding box co-ordinates and confidence scores)</p> <p>For scene classification task, the evaluation is done similarly as done in image classification and localization task where for each image an algorithm produces 5 scene categories organized in descending order based on confidence score which allows prediction of multiple labels for a single scene as human often refers to a scene using different names. The error is calculated by computing the distance between predicted scene label and actual scene label</p>

(continued)

**Table 2** (continued)

Year	Description of databases	Objective
2016	Datasets used for performing object detection, classification and localization, scene classification tasks are similar to the datasets used in previous year's challenges but the validation and the test sets used for performing object detection from video task are partially refreshed by adding new images. In addition to these tasks, scene parsing is also undertaken in ILSVRC 2016 challenge which is done using ADE20K dataset [7]	Evaluation of participating algorithms in ILSVRC 2016 challenge are done similarly for object localization, object detection, object detection from video, scene classification tasks are done similarly as done in ILSVRC 2015 challenge The main objective of the scene parsing task is to segment each image into different regions where each image is associated with semantic categories like road, sky, etc. and parsing through those regions The evaluation of the algorithms participating in the scene parsing task is done by computing two parameters namely, pixel-wise accuracy which represents the ratio of correctly predicted pixels and class-wise Intersection over Union (IoU) of pixels averaged over all 150 semantic categories
2017	Datasets used for object detection and object localization are similar to those used in previous challenges but for the object detection from video task, the validation and test sets are partially refreshed by adding new images	The evaluation of algorithms participating in object detection, object localization, and object detection from video tasks are evaluated similarly as done in previous challenges

### 2.3 Object Tracking Benchmark (OTB) (OTB100 [8], OTB 2013 [9], and OTB 2015 [10])

These popular databases are widely used to evaluate the performances of visual tracking algorithms. OTB benchmark has two versions namely, OTB 2013 and OTB 2015 which comprises of 51 sequences and 100 sequences, respectively. These databases contain annotated frames containing attributes belonging to a set of 11 challenging attributes for, e.g., illumination variation, motion blur, in-plane rotation, etc.

### 2.4 Common Objects in Context (COCO) (2014–2020) [11]

COCO database comprises of data which are used for bench-marking various computer vision tasks for, e.g., object detection, image segmentation, and image captioning. COCO database includes a total of 330 thousand images (out of which

200 thousand are labeled images) and 1.5 million object instances belonging to 80 object categories and 91 stuff categories and are mostly used for performing object segmentation and recognition in context tasks.

## 2.5 *Large-Scale Single Object Tracking (LaSOT) [12]*

This bench-marking database is created to train deep neural network-based object tracking methods which require huge amount of data for accurate training. So to fulfill their requirements, this database is designed comprising of 1400 sequences having more than 3.52 million high-quality, manually annotated frames belonging to 70 object categories where twenty sequences are included for each object category. The average length of videos included in this database is 83 s and the average number of frames present in the videos is 2512.

## 2.6 *UAV123 [13]*

Unlike all the databases whose details are discussed so far, UAV123 dataset is designed totally from aerial tracking viewpoint. It comprises of 123 video sequences captured from low-altitude Unmanned Aerial Vehicles (UAV). The inclusion of fully annotated 110 thousand frames and 123 video sequences in this database makes it one of the largest visual tracking database ranking only after Amsterdam Library of Ordinary Videos for tracking (ALOV++) ALOV300 ++ [14].

## 2.7 *Amsterdam Library of Ordinary Videos for Tracking (ALOV++) Database [14]*

This database is probably the largest database designed for bench-marking tracking algorithms. The videos included in this database are mostly short videos whose duration ranges between 9.2 and 35 s. In addition to these short videos, this database also contains 10 long videos whose duration ranges between 1 and 2 min.

Short videos containing various attributes like clutter, transparency, occlusion, etc. are mostly included in this database to maximize diversity.

## 2.8 Visual Object Tracking (VOT) (2013–2022) [15]

VOT challenges are organized every year from 2013 onwards to evaluate the performances of short-term trackers. These challenges also provide a global platform to researchers to discuss about the advancements conducted in this research area. Brief details of datasets used in VOT challenges conducted every year from 2013 onwards are given in Table 3.

Parameterized comparative analysis of existing object tracking databases are given in Table 4.

The details of the datasets discussed in this section are some of the most well-known datasets existing in this research area but none of these datasets are designed focusing on training self-driving vehicles which is one of the most popular application in object detection and tracking research area in recent years.

To solve this limitation in this work, we have designed a novel dataset namely, SDinIWTrack database comprising of thousands of annotated sequences of road videos captured during several inclement weather conditions as well as during varied illumination conditions. The videos whose sequences are included in this database are collected from several publicly available sources and are annotated by a group of annotators. Detailed description of the novel database are given in Sect. 4.

## 3 SDinIWTrack: A Novel Benchmark Database

In addition to the comprehensive survey conducted on existing object detection and tracking databases, the main contribution of this work lies in the creation of a novel database namely, SDinIWTrack database which comprises of thousands of annotated sequences containing target objects in mostly challenging visually degraded backgrounds to perform proper training of self-driving vehicles.

The road scene sequences which are included in SDinIWTrack database are obtained from publicly available sources and mostly contain target objects like pedestrian, traffic signals, vehicles, etc. in challenging conditions as the sequences are mostly captured during inclement weather conditions like snowy, rainy, foggy, etc. as well as during varied illumination conditions. These criteria are chosen in this work for selecting the videos as the fatal accidents occur mostly during inclement weather conditions and under poor illumination condition mostly during night.

Some examples of annotated frames included in the novel SDinIWTrack database are given in Fig. 1.

SDinIWTrack database is designed to perform effective training of self-driving vehicles to perform reliable detection of objects located within their close vicinity to avoid fatal accidents caused due to degraded visibility occurring during inclement weather and poor illumination conditions. Apart from self-driving vehicles, SDinIWTrack database can also be used to perform detection of objects by driver assistant systems installed in many vehicles for similar purpose as according to the recent

**Table 3** Brief details of datasets used in VOT challenges

Year	Dataset
2013	The dataset used in this challenge comprises of 16 short sequences containing varied objects and challenging backgrounds which are selected from a large pool of sequences using a clustering method which selects these videos based on the characteristics of objects and backgrounds. The sequences are manually annotated using axis-aligned bounding boxes
2014	The dataset used in this challenge comprises of 25 short sequences containing varied objects and challenging backgrounds out of which 8 sequences for, e.g., bicycle, diving, etc. are included from VOT 2013 dataset. These sequences are selected using the same clustering method as done in VOT 2013 challenge. The new sequences contain complementary backgrounds and objects and are manually annotated with rotated bounding boxes which facilitates production of highly accurate ground truths
2015	This dataset contains 60 short sequences which were selected from a large set of sequences included in several well-known datasets whose details are given in [16] using clustering method. These sequences are manually annotated using rotated bounding boxes which were generated by a group of annotators and are checked by other two groups of annotators and are finally checked by co-ordinator of the annotation process. After final checking, if the ratio between the shortest and longest side of any bounding box side is found to be exceeding 0.95, then the rotated bounding box is replaced with axis-aligned bounding box
2016	The dataset used in VOT 2016 challenge is similar to the dataset used in VOT 2015 challenge but the sequences are re-annotated in this year's challenge to provide more accurate ground truths. In addition to the main VOT 2016 challenge, a sub-challenge named VOT-TIR 2016 is also co-organized focusing on thermal imagery tracking. The dataset created for this sub-challenge comprises of new sequences
2017	Like VOT 2016 challenge, in 2017 also a sub-challenge namely VOT-TIR 2017 is organized alongside VOT 2017 main challenge. The dataset used in VOT-TIR 2017 challenge is similar to the dataset used in VOT-TIR 2016 sub-challenge
2018	Like VOT 2016 and VOT 2017 challenges, VOT 2018 challenge also co-hosted a sub-challenge focusing on long-term tracking namely, VOT-LT2018. All the sequences included in these datasets are manually annotated with rotated bounding boxes
2019	VOT 2019 hosted a group of challenges which are listed as follows: <ul style="list-style-type: none"> <li>• VOT-ST2019 and VOT-RT2019 which are focused on performing robust short-term tracking under different challenging conditions like occlusion, appearance variation, etc. and time constraints, respectively</li> <li>• VOT-LT2019 which is focused on performing robust long-term tracking with target disappearance</li> <li>• VOT-RGBT2019 and VOT-RGBD2019 which are focused on performing infrared and thermal short-term tracking challenge and color and depth long-term tracking challenge using four channels (RGB + IR) and four channels (RGB + depth), respectively</li> </ul>
2020	VOT 2020 hosts five challenges namely, VOT-ST2020, VOT-RT2020, VOT-LT2020, VOT-RGBT2020, and VOT-RGBD2020 which have similar objectives as of their counterparts in 2019. All the tool-kits used in these challenges are re-implemented using Python in 2020
2021	VOT 2021 challenge co-hosted four challenges namely, VOT-ST2021, VOT-RT2021, VOT-LT2021, and VOT-RGBD2021

(continued)

**Table 3** (continued)

Year	Dataset
2022	<p>VOT 2022 co-hosted seven challenges as listed below:</p> <ul style="list-style-type: none"> <li>• VOT-ST2022 and VOT-RTS2022 challenges which are focused on performing real-time, short-term tracking of objects contained in input RGB images localized by segmentation masks</li> <li>• VOT-STB2022 and VOT-RTB2022 challenges which are focused on performing real-time, short-term tracking of objects contained in input RGB images localized by bounding boxes</li> <li>• VOT-RGBD2022 tracking challenge which has similar objective as VOT-RGBD 2021</li> <li>• VOT-D2022 and VOT-LT2022 are focused on performing short-term and long-term tracking of target objects contained in RGB images localized by bounding boxes</li> </ul>

**Table 4** Parameterized comparative analysis of existing object tracking database

Database	Database content	Number of object categories	Objective
PASCAL VOC 2012 [1]	6929 annotated objects training/validation set: 27,450 images	20 object categories	Classification detection
ILSVRC 2017 [4]	Training set: 10,000,000 hand-labeled images, Test/validation: 150,000 images	Localization: 1000 object categories Detection: 200 fully labeled object categories Detection from videos: 30 fully labeled object categories	Localization detection
OTB 2013 [9]	51 sequences	11 attributes	Detection
OTB 2015 [10]	100 sequences	11 attributes	Detection
COCO 2020 [11]	300 thousand image frames with approximately 1.5 million object instances	80 object categories 91 stuff categories	Segmentation and recognition
LaSOT [12]	1400 sequences with 3.52 million high-quality manually annotated frames	70 object categories	Tracking
UAV123 [13]	123 video sequences with more than fully annotated 110 K frames	Not mentioned	Aerial tracking
ALOV++ [14]	315 video sequences with 89,364 annotated boxes	64 object categories	Segmentation tracking

**Fig. 1** Examples of annotated sequences included in SDinIWTrack database



reports published in [17–20], it have been stated that every year a large number of people dies or gets injured due to fatal accidents occurring due to the mentioned causes.

For this purpose, thousands of frames of road scenes possessing varied illumination characteristics and different extents of visibility degradation obtained from videos collected from publicly available sources are included in the designed database. The road scenes possessing various degrees of degradation and challenging conditions are included in the designed database to perform effective training of self-driving vehicles or driver assistance systems in a varsity of challenging situations.

To the best of our knowledge, SDinIWTrack is the first database which is designed focusing on performing effective training of self-driving vehicles and driver assistance systems, no other existing databases whose details are discussed in Sect. 2 are designed with this objective.

SDinIWTrack database mostly possess road scenes containing objects belonging to categories like pedestrian, vehicles, trees, traffic lights, etc. but there may be numerous other objects belonging to different categories which can be encountered while driving, then in such cases, the systems trained using instances belonging to SDinIWTrack database cannot detect such objects satisfactorily.

## 4 Conclusion

The main contribution of the work lies in the creation of a novel dataset namely, SDinIWTrack database which comprises of thousands of annotated sequences of road scenes captured during inclement weather conditions like rainy, foggy, sunny, etc. and under varied illumination conditions to perform proper training of self-driving vehicles to avoid fatal accidents. The main objective of this work is to create a proper dataset including annotated sequences of road scenes containing target objects for, e.g., pedestrian, vehicles, traffic signals, etc. in challenging conditions focusing mainly on training self-driving vehicles and driver assistance systems owing due to the relevance of this application in recent years as to the best of our knowledge, none of the existing databases in this research area are created with this objective. We hope that SDinIWTrack will have significant contribution in this research area and will help researchers to perform proper training of self-driving vehicles and driver assistance systems in future as it includes various challenging cases.

## References

1. <http://host.robots.ox.ac.uk/pascal/VOC/>. Retrieved 26 Jan 2023
2. <https://www.kaggle.com/datasets/hsankesara/flickr-image-dataset>. Retrieved 26 Jan 2023
3. <https://www.microsoft.com/en-us/download/details.aspx?id=52644>. Retrieved 26 Mar 2023
4. Russakovsky O, Deng J, Su H, Krause J et al (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vision* 115:211–252
5. Deng J, Dong W, Socher R, Li L-J et.al (2009) ImageNet: a large-scale hierarchical image database. in IEEE computer vision and pattern recognition (CVPR), Miami, FL, USA, pp 248–255
6. <https://www.kaggle.com/datasets/nickj26/places2-mit-dataset>. Retrieved 08 Feb 2023
7. <https://groups.csail.mit.edu/vision/datasets/ADE20K/>. Retrieved 08 Feb 2023
8. Wu Y, Lim J, Yang M-H (2015) Object tracking benchmark. *IEEE Trans Pattern Anal Mach Intell* 37(9):1834–1848
9. Wu Y, Lim J, Yang M-H (2013) Online object tracking: a benchmark. In: IEEE conference on computer vision and pattern recognition (CVPR), USA
10. <https://paperswithcode.com/dataset/otb-2015>. Retrieved 08 Feb 2023
11. <https://cocodataset.org/#home>. Retrieved 08 Feb 2023
12. Fan H, Lin L, Yang F et al (2019) LaSOT: a high-quality benchmark for large-scale single object tracking. In: IEEE computer vision and pattern recognition (CVPR), Long Beach, CA
13. Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for UAV tracking. In European conference on computer vision (ECCV), Amsterdam, pp 340–353
14. Smeulders AWM, Chu DM, Cucchiara R et al (2014) Visual tracking: an experimental survey. *IEEE Trans Pattern Anal Mach Intell (PAMI)* 36(7):1442–1468
15. <https://www.votchallenge.net/>. Retrieved 08 Feb 2023
16. <https://www.votchallenge.net/vot2015/dataset.html>. Retrieved 08 Feb 2023
17. <https://www.hindustantimes.com/india-news/cold-wave-grips-north-india-3-killed-40-injured-in-fog-related-accidents101671556602987.html>. Retrieved 24 Jan 2023
18. <https://www.dailypioneer.com/2022/pioneer-exclusive/dense-fog-shrouds-north-india.html>. Retrieved 22 Jan 2023

19. <https://www.ndtv.com/india-news/heavy-fog-over-north-india-low-visibility-causing-motor-accidents-3631375>. Retrieved 21 Jan 2023
20. <https://www.cartoq.com/more-than-25-vehicles-collide-on-yamuna-nagar-highway-due-to-dense-fog-video/>. Retrieved 24 Jan 2023

# Detection of Cipher Types Using Machine Learning Techniques



Abhiroop Mukherjee, Arnab Sen, Krishnendu Bera, Rajdeep Ghosh,  
Swarnali Mondal, Sanjana Chakravarty, Subinoy Sikdar, and Malay Kule

**Abstract** The identification of a cryptosystem has been a challenge for decades. This paper's main objective is to identify the type of cryptosystem used to encrypt a particular text. We have explored the realm of machine learning to recognize a pattern among complex classical ciphertexts that generally have a simple representation in plaintext. We have modeled our objective as a sequence-to-sequence learning task that we have tried to solve using Convolution Neural Networks (CNNs) and state-of-the-art Transformer models. With only a tiny dataset (130k) consisting of ciphertexts and the corresponding cryptosystem used to encrypt the same, our model has shown a good accuracy of 96.72 % which proves a significantly steep learning curve compared to other sequence-to-sequence models. Here we show the enormous potential of these models and how they can perform even better if the barrier of resources and computation time is lifted.

**Keywords** Machine learning (ML) · Cryptosystem identification · Classical ciphertext classification · Convolution neural networks (CNNs) · Transformer models

## 1 Introduction

In today's world, information is a powerful tool. However, it is essential to protect information and data from unauthorized access, distortion, or any other unauthorized activity. Cryptography is the study of secret writing and is used to prevent unauthorized access to data. It works by using a key(s) to convert plaintext messages into unreadable ciphertext and vice versa. Encryption is the process of converting plaintext messages into an unreadable format to protect sensitive information (called ciphertext), while the inverse process is called decryption. Only authorized users can encrypt or decrypt messages [1].

---

A. Mukherjee · A. Sen · K. Bera (✉) · R. Ghosh · S. Mondal · S. Chakravarty · S. Sikdar · M. Kule  
Department of Computer Science and Technology, Indian Institute of Engineering Science and  
Technology, Shibpur, West Bengal 711103, India  
e-mail: [bera.krishnendu36@gmail.com](mailto:bera.krishnendu36@gmail.com)

Previous studies on detecting the type of ciphers have used ANN, transformer, ensemble model of feature-engineering, and BLSTM-GRU cell unit network. The research, which explored the use of Long Short-Term Memory (LSTM) and Transformer algorithms as feature-learning algorithms for the task of identifying 55 classical cipher types, reported that they achieved an accuracy of 82.78% over a dataset of 10 million ciphertexts with an ensemble model that combined feature-engineering [2] and feature-learning neural network types. One more work presented the progress of a research project that uses a TensorFlow-based network [3] for detecting the cipher type. The network, in this case, could classify about 90% of the ciphertexts in a publicly available challenge correctly, and their method resulted in a saving of about 54% computation time compared to using different solvers. Another study based on deep Bidirectional LSTM-Gated Recurrent Units (GRU) [4] for ciphertext classification evaluated the model on two publicly available datasets and reported a high classification accuracy of 95.8%.

The goal of our work is to use artificial neural networks to examine encrypted text files and classify them based on their cipher types. The process involves creating a collection of ciphertexts with various cipher types, building a deep neural network (DNN), and training it with a variety of ciphertexts and training parameters to enable accurate learning and ciphertext classification [3]. Once the DNN is trained, it can be used to determine the type of ciphertext. In this work, the Caesar, Affine, Vigenère, Substitution, and Rail Fence ciphers were used to encrypt English texts. The encrypted files were then submitted to CNN and transformer models, which made predictions, and the accuracy of these predictions was assessed. Our approach presents an improvement of 14% compared to them ensemble model of feature-engineering, 6% improvement compared to ANN and 1.5% improvement compared to the BLSTM-GRU network model. It is important to note that our results not only show the potential of these models but also highlight the possibility of even better performance if the limitations of resources and computation time are addressed.

This paper is organized into five different sections. Section 2 describes the preliminaries for this work followed by this introductory section. Section 3 elaborates on our proposed work with dataset generation and a discussion of the models. Section 4 shows the experimental findings. Section 5 contains the conclusion and the work that we plan to address in the future.

## 2 Preliminaries

This paper requires basic knowledge of neural networks and deep learning, as well as an overview of the encryption algorithms used. These algorithms can be useful for feature extraction. The main machine learning topics covered are artificial neural networks (ANNs), perceptrons, recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers.

## 2.1 Artificial Neural Network (ANN)

An artificial neural network (ANN) is a type of computational model that mimics the human brain structurally and functionally [5]. ANNs consist of layers of interconnected neurons, which process input data and generate output values. ANNs are trained to perform numerous types of tasks, such as classification, regression, and clustering.

## 2.2 Perceptron

A perceptron is a type of artificial neural network (ANN) that can be used for binary classification tasks [6]. It consists of a single layer of neurons, with each neuron receiving input from multiple sources, and producing a single output value. A set of weights determines the output of a perceptron. Those weights are then applied to the input values, and a threshold value is used to determine whether the output is 0 or 1.

## 2.3 Convolutional Neural Network (CNN)

CNNs consist of multiple layers of interconnected neurons, which process the input data in a hierarchical manner [7]. The first layer of a CNN typically consists of a set of filters. These filters are used to extract features such as edges and textures by applying them to the input data. These features are then passed to the next layer of the CNN, where they are combined and processed to extract more complex features, such as shapes and objects. The output of a CNN is a set of probabilities that indicate the likelihood that the input data belongs to each of the possible classes.

## 2.4 Transformers

The transformer is a type of encoder-decoder model, which is commonly used for tasks such as machine translation and text summarization [8]. The use of self-attention, a mechanism that enables the model to learn relationships between different words in the input text, is the transformer's key innovation. This allows the transformer to process the input sentence in parallel, rather than in a sequential manner, which makes it much more efficient and scalable than previous NLP models.

## 2.5 Attention Mechanism

Self-attention is a mechanism used in the transformer [8]. With self-attention the model learns dependencies between different words in the input, by computing weighted sum of the input words based on their relevance to each other.

## 3 Proposed Work

The focus is on generating a representative dataset and developing a CNN-based as well as a transformer-based model for the task. The goal is to achieve high levels of accuracy and robustness through training and validation.

### 3.1 Dataset Generation

The dataset has been generated for five types of ciphers:

- Caesar
- Affine
- Vigenère
- Substitution
- Rail Fence

The plaintext used for encryption was the texts from Shakespeare’s Hamlet [9], Macbeth [10], Merchant of Venice [11] and Romeo and Juliet [12].

We converted the text to lowercase and removed all whitespaces and punctuation. Then the plaintext was encrypted and exported as a CSV.

### 3.2 CNN Based Proposed Model

The main intuition behind this method is that CNN uses its kernel as a “local receptive field” [7], which means that a CNN kernel (small in size as we are dealing with stream cipher and not block cipher) can look its nearby characters and will find some pattern in the ciphertext.

So, we first converted every ciphertext into their respective list of tokens using a TextVectorizer [13], and then we converted that list into an image as shown in Fig. 1.

Note that this conversion depends on the length of the string used, i.e. for a string of length  $n$ , we will generate a  $n \times n$  matrix. After we generate all variable-sized matrices, we resize them to  $64 \times 64$  using Bilinear Interpolation.

**Fig. 1** Converting text to 2D text matrix

$$\begin{bmatrix} a & b & c & d \end{bmatrix} \longrightarrow \begin{bmatrix} a & b & c & d \\ b & c & d & a \\ c & d & a & b \\ d & a & b & c \end{bmatrix}$$

Now we can see that a kernel (say  $3 \times 3$  with stride one) can observe its previous as well as its next two characters and can use that information to find patterns in the ciphertext.

We then use a standard CNN model [7] to classify these images and compare them with a fine-tuned ConvNeXt [14] Model.

### 3.3 Motivation Behind Choosing Transformer Model

In our case, we used pre-trained transformer models, which have been fine-tuned using our own dataset. This allowed us to leverage the powerful language understanding capabilities of the transformer architecture while also ensuring that the model was specifically tailored to our problem.

The transformer architecture is based on the idea of self-attention, which allows the model to weigh the importance of different tokens in the input sequence when making predictions [8]. This is done by computing a dot product between the input tokens and a set of learnable parameters, which are then used to generate a weight for each token. These weights are then used to compute a weighted average of the input tokens, which is used as the input to the next layer of the model.

This self-attention mechanism allows the transformer to effectively capture long-range dependencies between tokens, which is crucial for understanding the relationship between the ciphertext and the algorithm used to generate it.

In summary, we have used the transformer model as it is a state-of-the-art model for sequence-to-sequence problems and its self-attention mechanism and multi-head attention allow the model to capture long-range dependencies between tokens and attend to different parts of the input sequence in parallel, which is crucial for understanding the relationship between the ciphertext and the algorithm used to generate it.

### 3.4 Transformer-Based Proposed Model

The Transformer in NLP is a novel architecture that aims to solve sequence-to-sequence tasks while efficiently handling long-range dependencies. The Transformer was proposed in the paper “Attention Is All You Need” [8]. The main intuition behind this method is to compute the relative importance of a ciphertext character with respect to the other ciphertext characters.

Our dataset contains encrypted text and the algorithm used for encryption, we create a text prompt from each row for example:

Ciphertext: `ymccaygimgamkyyiycuycqocyeagmigkycmyyooyy`

Algorithm: `affine`

The model learns the relative importance of each token (character) and uses it to predict the type of ciphertext. For instance, if the model is trained using the previous prompt and we then provide the following prompt:

Ciphertext: `yycmdkreqluithnalekstndonwothvn`

Algorithm:

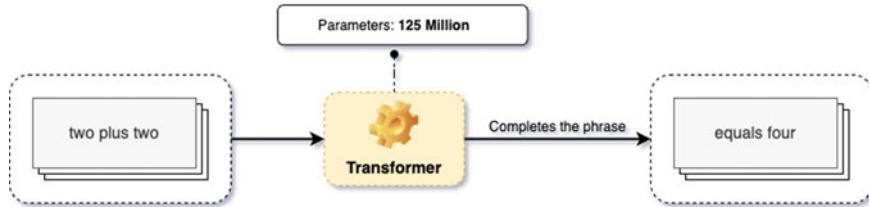
The model may generate the output as “affine” which it learned from training.

In our use-case, we have used the GPT-Neo (Generative Pre-Trained) [15] and OPT [16] models both with 125 million parameters to achieve our objective. GPT-Neo has been pre-trained on the Pile [17] dataset created by EleutherAI, whereas the OPT was trained on the Pile [17] as well as on The Pushshift Reddit Dataset [18]. The GPT-Neo architecture is an extension to GPT2 [19] using local attention in every other layer with a window size of 256 tokens. On the other hand, the OPT’s design is based on the latest GPT-3 [20] and both OPT and GPT-3 have a comparable performance. All these models are available for free in the Hugging Face [21] transformers library which can be used for a number of NLP tasks. The following steps were performed:

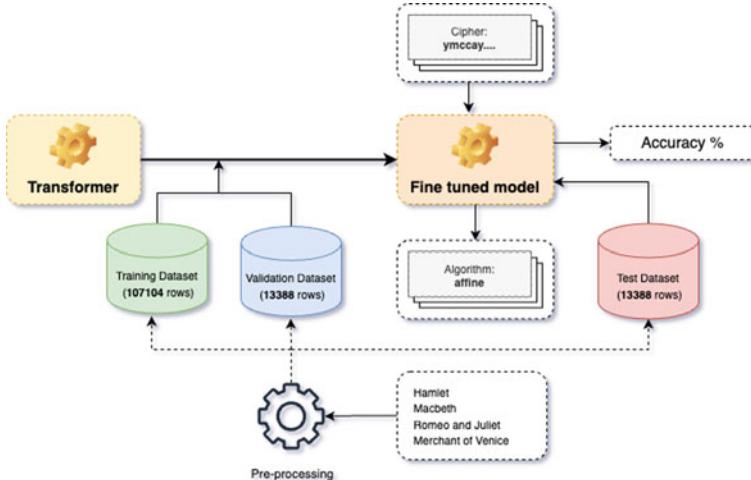
1. Each data is converted into text prompts that is passed through a tokenizer that creates embedding for each token in our input string.
2. The input dataset has been split into the ratio 80:10:10 for training, testing, and validating purposes.
3. The training and validation set has been used to fine-tune the model’s hyper-parameters specific to our use-case, i.e., detecting the algorithm used in the ciphertext generation.
4. We used the fine-tuned model on our test dataset to check the accuracy of our model.
5. After fine-tuning, for each of our test data we performed sampling with the temperature set to 0.7 to return 10 predictions. The maximum predicted algorithm is chosen to be the correct one and is then compared with the actual value to compute the accuracy.

For our testing purpose, we used various decoding strategies to find the one which gives the best result. Out of the strategies: Greedy Search, Beam Search, and Sampling with set temperature, we found sampling to give the most accurate results of all three.

The greedy method selects the next token that has the highest probability as the output while sampling selects one of the possible next tokens randomly. When we set a temperature, the randomness is lowered such that the likelihood of high-probability words is increased and that of the low probability is decreased. We found that a temperature set to 0.7 gave the optimum accuracy on our test data. 10 predictions are computed for each test data and the maximum predicted algorithm is chosen to be the correct one and is then compared with the actual value to compute the accuracy.



**Fig. 2** Pre-training of a transformer model on a large corpus



**Fig. 3** Fine-tuning of the pre-trained transformer model on a custom dataset

Figure 2 depicts the pre-training phase of a transformer-based architecture. Upon completion of this phase, the model has acquired the capability of generating generic text. As an illustration, it can predict the continuation of the phrase “two plus two” as “equals four”.

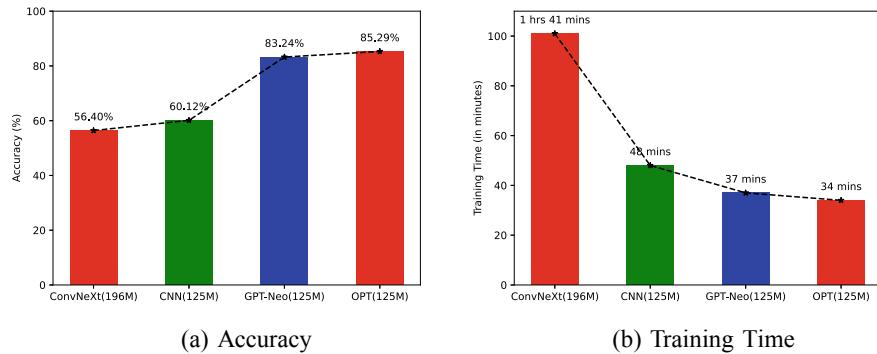
Figure 3 represents the pre-trained transformer-based model that has undergone fine-tuning utilizing a custom dataset derived from several works of Shakespeare. The fine-tuned model is now capable of being applied to the specific task of identifying the cipher type of a given ciphertext.

## 4 Experimental Results

This report presents the findings of experiments conducted on two datasets to evaluate the performance of a specific model or approach, including accuracy, robustness, and generalizability. The results have been presented in a concise manner, highlighting significant trends, patterns, or insights.

**Table 1** Training time and accuracy on dataset-1

Model	Number of classes	Training time (mins)	Accuracy (%)
ConvNeXt(196M)	5	1 h 41	56.40
CNN(125M)	5	48	60.12
GPT-Neo(125M)	5	37	83.24
OPT(125M)	5	34	85.29

**Fig. 4** Comparison of accuracies and training times on dataset-1

## 4.1 Dataset-1

Our initial dataset consisted of 34,075 rows, which were split into training, testing, and validation sets in a ratio of 80:10:10. The sizes of each split are as follows:

- Train dataset: 27,259 rows
- Validation dataset: 3408 rows
- Test dataset: 3408 rows.

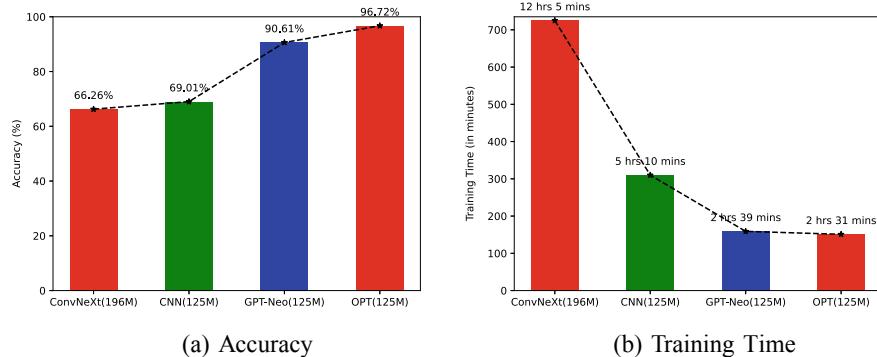
Table 1 shows the results obtained using the dataset-1. We observed that training a CNN model from scratch gives a better result than fine-tuning a pre-trained CNN because pre-trained CNN models are trained on object classification, which differs significantly from our ciphertext classification problem. We also observed that transformer-based models like GPT-Neo and OPT performed much better than CNN-based models. The result of the table are shown as BarPlot in Fig. 4.

## 4.2 Dataset-2

Next, we increased our dataset, which now contains 133,880 rows. This dataset was split into training, testing, and validation sets in a ratio of 80:10:10. The sizes of each split are as follows:

**Table 2** Training time and accuracy on dataset-2

Model	Number of classes	Training time	Accuracy (%)
ConvNeXt(196M)	5	12 h 5 mins	66.26
CNN(125M)	5	5 h 10 mins	69.01
GPT-Neo(125M)	5	2 h 39 mins	90.61
OPT(125M)	5	2 h 31 mins	96.72

**Fig. 5** Comparison of accuracies and training times on dataset 2

- Train dataset: 107,104 rows
- Validation dataset: 13,388 rows
- Test dataset: 13,388 rows.

Table 2 shows the results obtained using dataset-2. Figure 5 shows that the results obtained using OPT and GPT models are far better than other models based on CNN.

## 5 Conclusion

This paper illustrates the current progress of our work in detecting ciphertext using Artificial Neural Networks. Here, we used three types of neural networks to achieve our objective: Convolutional Neural Networks, GPT-Neo, and OPT.

With the CNN approach, our model was able to achieve an accuracy of 68.30% for 5-class classifications. We saw a significant improvement with the transformers approach using GPT-Neo on the same dataset and classes, giving an accuracy of 90.61%. This was primarily due to the ability of GPT-Neo to memorize the local importance of tokens due to the self-attention feature. OPT gave better improvements, giving an accuracy of 96.72%, signifying that OPT is better than GPT-Neo for the

task of detecting the type of ciphertext. We also observed that transformer-based models are giving better results than CNN while taking less time for training.

Our work can be further extended using similar transformer-based models like BERT, GPTJ, and Bloom to see if those models are better suited for the task. We will also increase the number of classes of ciphertext and the size of the dataset to see if we observe any improvements.

## References

1. Katz J, Lindell Y Introduction to modern cryptography. [http://staff.ustc.edu.cn/~mfy/moderncrypto/reading%20materials/Introduction\\_to\\_Modern\\_Cryptography.pdf](http://staff.ustc.edu.cn/~mfy/moderncrypto/reading%20materials/Introduction_to_Modern_Cryptography.pdf)
2. Leierzopf E, Mikhalev V, Kopal N, Esslinger B, Lampesberger H, Hermann E (2021) Detection of classical cipher types with feature-learning approaches. [https://link.springer.com/chapter/10.1007/978-981-16-8531-6\\_11](https://link.springer.com/chapter/10.1007/978-981-16-8531-6_11)
3. Kopal N (2020) Of ciphers and neurons—detecting the type of ciphers using artificial neural networks. [https://www.researchgate.net/publication/341517754\\_Of\\_Ciphers\\_and\\_Neurons\\_-\\_Detecting\\_the\\_Type\\_of\\_Ciphers\\_Using\\_Artificial\\_Neural\\_Networks](https://www.researchgate.net/publication/341517754_Of_Ciphers_and_Neurons_-_Detecting_the_Type_of_Ciphers_Using_Artificial_Neural_Networks)
4. Ahmadzadeh E, Kim H, Jeong O, Kim N, Moon I A deep bidirectional lstm-gru network model for automated ciphertext classification. <https://ieeexplore.ieee.org/document/9668927>
5. Zupan J (1994) Introduction to artificial neural network (ANN) methods: what they are and how to use them. Acta Chimica Slovenica 41:327–327
6. Gallant SI et al (1990) Perceptron-based learning algorithms. IEEE Trans Neural Netw 1(2):179–191
7. O’Shea K, Nash R (2015) An introduction to convolutional neural networks. <http://arxiv.org/abs/1511.08458>
8. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. <https://arxiv.org/abs/1706.03762>
9. Shakespeare W (2015) 2015/hamlet.txt at master . cs109/2015. <https://github.com/cs109/2015/blob/master/Lectures/Lecture15b/sparklect/shakes/hamlet.txt>. Accessed on 11 Mar 2023
10. Shakespeare W (2015) 2015/macbeth.txt at master . cs109/2015. <https://github.com/cs109/2015/blob/master/Lectures/Lecture15b/sparklect/shakes/macbeth.txt>, Accessed on 09 Mar 2023
11. Shakespeare W (2015) 2015/merchantofvenice.txt at master . cs109/2015. <https://github.com/cs109/2015/blob/master/Lectures/Lecture15b/sparklect/shakes/merchantofvenice.txt>. Accessed on 15 Mar 2023
12. Shakespeare W (2015) 2015/romeojuliet.txt at master . cs109/2015. <https://github.com/cs109/2015/blob/master/Lectures/Lecture15b/sparklect/shakes/romeojuliet.txt>. Accessed on 04 Mar 2023
13. Tensorflow: tf.keras.layers.textvectorization | tensorflow v2.11.0. [https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/TextVectorization](https://www.tensorflow.org/api_docs/python/tf/keras/layers/TextVectorization). Accessed on 25 Mar 2023
14. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. <https://arxiv.org/abs/2201.03545>
15. Neo G (2023) Gpt neo. [https://huggingface.co/docs/transformers/model\\_doc/gpt\\_neo](https://huggingface.co/docs/transformers/model_doc/gpt_neo). Accessed on 17 Feb 2023
16. Zhang S, Roller S, NGMAMCSCCD et al (2022) Open pre-trained transformer language models. <https://doi.org/10.48550/arXiv.2205.01068>
17. Gao L, Biderman S, Black S, Golding L, Hoppe T, Foster et al The pile: an 800gb dataset of diverse text for language modeling. <https://arxiv.org/abs/2101.00027> (2020)
18. Baumgartner J, Zannettou S, Keegan B, Squire M, Blackburn J (2020) The pushshift reddit dataset. <https://arxiv.org/abs/2001.08435>

19. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I et al Language models are unsupervised multitask learners. OpenAI blog
20. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S (2020) Language models are few-shot learners. <https://doi.org/10.48550/arXiv.2005.14165>
21. Face H (2023) Hugging face—the ai community building the future. <https://huggingface.co/>. Accessed on 17 Feb 2023

# An Overview of Segmentation Models for the Extraction of Brain Tissues from Magnetic Resonance Images



Elisabeth Thomas and S. N. Kumar

**Abstract** Image segmentation plays an inevitable role in biomedical image processing for the delineation of anatomical organs and affected tissues. The main focus of this research work is the analysis of algorithms to segment the tissues in MR brain images, which aids in the diagnosis of neuro disorders. The widely used segmentation algorithms for the extraction of brain tissues are thresholding, clustering, atlas-guided and deep learning models. A qualitative study of widely used segmentation approaches for the MR brain images is discussed in this chapter. In the current scenario, hybrid segmentation approach gains prominence in the extraction of brain tissues from MR images. The simulation results of the expectation–maximization algorithm for the segmentation of brain tissues from MR images are also furnished in this chapter and validated by performance metrics.

**Keywords** White matter (WM) · Grey matter (GM) · Cerebrospinal fluid (CSF) · Expectation–maximization algorithm · Segmentation

## 1 Introduction

Image segmentation is the method of partitioning an image into distinct and non-overlapping regions that share similar characteristics such as color, depth, texture, or intensity. This technique provides a way to label images with similar regions or create boundaries around them using contours. In medical research, MRI image segmentation is an essential tool that provides guidelines for treating various neurological disorders. Various methods, including automatic image segmentation techniques, are used to segment MRI images into WM, GM and CSF to identify morphological

---

E. Thomas  
Lincoln University College, 15050 Kota Bharu, Malaysia  
e-mail: [ethomas@lincoln.edu.my](mailto:ethomas@lincoln.edu.my)

S. N. Kumar (✉)  
Department of EEE, Amal Jyothi College of Engineering, Kottayam, Kerala 686518, India  
e-mail: [appu123kumar@gmail.com](mailto:appu123kumar@gmail.com)

changes associated with neurological disorders [1, 2]. Utilizing a magnetic field, radio waves, and a computer, magnetic resonance imaging (MRI) is a diagnostic tool that produces detailed images of internal organs and structures within the body, without the need for invasive procedures. It is widely used to diagnose and monitor different conditions affecting the brain, spine, joints, and tumors or injuries. MRI is a safe imaging choice for many patients because it does not use ionizing radiation [3].

White matter consists of nerve fibers (axons) that link different regions of the brain and spinal cord, which is enveloped by a fatty substance known as myelin, giving it its characteristic color. Its primary role is to transmit signals between various regions of the brain and spinal cord. Grey matter, on the other hand, is composed of nerve cell bodies, dendrites, and synapses, and is responsible for information processing and movement control. It is present in the brain's cortex, as well as other regions such as the cerebellum and brainstem. Cerebrospinal fluid (CSF), a colorless liquid that surrounds the brain and spinal cord, is produced in the brain's ventricles and circulates throughout, providing nutrients, oxygen, and waste removal, as well as cushioning and pressure regulation [3, 4].

WM, GM, and CSF [4, 5] are all vital components of the brain and spinal cord, and their extraction can be beneficial in various applications, such as research, medical diagnosis, and treatment planning. For instance, in research, their extraction can aid in studying brain structure and function, identifying disease biomarkers, and developing new therapies. In medical diagnosis, their extraction can help detect brain abnormalities or injuries like strokes and traumatic brain injuries. In treatment planning, their extraction can assist in guiding the placement of surgical instruments or radiation beams and monitoring the brain's response to treatment over time [6].

Segmentation of WM, GM, and CSF in magnetic resonance (MR) images was performed using a variety of techniques, including manual segmentation, semi-automated methods, and fully automated methods. Some popular methods include thresholding, atlas-based segmentation, region growing, and machine learning techniques such as deep learning. Atlas-based segmentation uses a pre-existing “atlas” of the brain, which is a map of the brain's structure to segment the MRI [7, 8]. The atlas is registered, and the segmentation is done by transferring the labels from the atlas to the subject imaging data. The region-growing method starts with a seed point in the MRI and then groups together voxels that are similar in intensity to the seed point. The region continues to grow until all the pixels in the desired structure have been included. Deep learning-based segmentation uses deep neural networks to segment the MRI [9, 10]. They are trained on a large data set of labeled MRIs and can segment the MRI in real time. The choice of method will depend on the specific application and the availability of training data.

## 2 Related Works in the Segmentation of MR Brain Images

Enormous segmentation algorithms are there in image processing for the extraction of the wanted region of interest from an image, and the following algorithms are widely used for the segmentation of brain regions.

### 2.1 *Thresholding- and Clustering-Based Segmentation Models*

Thresholding is a widely used segmentation algorithm in computer vision and image processing. Bi-level thresholding is commonly used for simple regions of interest, while multilevel thresholding is used for complex regions. The Otsu technique-based multilevel thresholding was coupled with the improved whale optimization algorithm for image segmentation [11], while the advanced wind-driven optimization (AWDO) optimization [12] and opposition-based learning [13] hybrid rice optimization (OHRO) were coupled with the multilevel thresholding algorithm for the segmentation of brain tissues. In addition, the entropy-based multilevel thresholding was coupled with the African vulture optimization algorithm (OAVOA) [14] for the segmentation of brain and dermoscopic images. In [15], the role of nature-inspired optimization algorithms in the multilevel thresholding segmentation of images is highlighted.

Clustering algorithms such as GMM, mean-shift, HC, K-means, and FCM [16] were tested on MR brain images and authenticated by validation metrics in another study. FCM exhibits efficient results when compared to other classical clustering algorithms. Another brain segmentation method proposed an integrated neuro fuzzy  $k$ -means radial basis function (RBF) approach, which extracted white matter, gray matter, and CSF and was evaluated in terms of the misclassification rate and percentage of clustering [17]. This method significantly reduced the misclassification rate and had a high percentage of clustering compared to other algorithms like fuzzy logic,  $k$ -means, RBF, and fuzzy  $k$ -means algorithms [17]. Prior to the region of interest extraction, most segmentation algorithms carry out pre-processing. An improved FCM based on double estimation incorporating both original and denoised images was proposed in [18]. The segmentation results outperforms the results of FCM with single estimation [18].

## 2.2 *Atlas-Based and Probability-Theory-Based Segmentation Models*

The process of automatically extracting a region of interest from medical images is called atlas-based segmentation. This technique uses a pre-segmented image as a reference to segment various structures of interest, such as WM, GM, and CSF. One example of this is the FLAWS sequence, which is effective at segmenting subcortical gray matter and is fast, making it applicable to two distinct age ranges [19]. However, classical atlas-based segmentation is time-consuming, so a less computationally complex alternative, the One-pass Aligned Atlas Set for Images Segmentation (OASIS), was proposed in [20]. This method uses a deep learning model with the region of interest-based registration. Additionally, atlas-based segmentation can extract multiple regions of interest simultaneously. Validation studies have been done on MR brain images to evaluate this method [21]. The atlas-based model, along with the graph cut model, has been utilized for the segmentation of the hippocampus in MR brain images [22]. Monte Carlo algorithms are a type of computational method that uses random sampling and has gained prominence in solving real-world problems. In medical image segmentation, Monte Carlo algorithms have been used to segment brain tissues in MRI images. The algorithm estimates the probability distribution of tissue types in the image using random sampling, which is then used to segment the image into the desired tissues [23]. The expectation–maximization algorithm is a probabilistic model that is widely used in medical image segmentation, e.g., the modified expectation–maximization algorithm was devised for the segmentation of MR brain images [24]. An improved version of this algorithm was proposed in [25] for the segmentation of MR brain Alzheimer’s images. For classification, a logistic regression classifier was utilized, with feature extraction done by grey-level co-occurrence matrix and feature optimization by principal component analysis.

## 2.3 *Deep Learning-Based Segmentation Models*

Deep learning techniques for segmentation refer to a type of machine learning method that leverages artificial neural networks to partition images into distinct regions of interest. These methods have gained popularity in recent times, thanks to advancements in deep learning algorithms and the abundance of annotated training data. Through the application of various deep learning algorithms, such as B-UNET [26], brain tissues can be segmented in 3D brain MR images, aiding in the diagnosis and treatment of brain disorders. Several studies, including [27] and [28], have explored the use of hybrid segmentation models that combine deep neural networks and Gaussian mixture models for 3D brain tissue segmentation. Furthermore, parallel computing technology has been used [29] to implement deep learning models for brain tissue segmentation. The patch wise M net convolution neural network architecture was proposed in [30] for the segmentation of brain tissues.

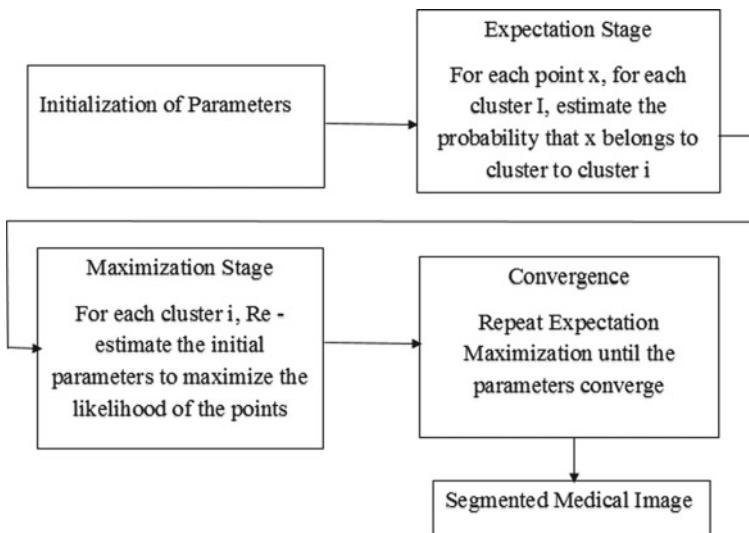
### 3 Expectation–Maximization (EM) Algorithm for the Segmentation of Brain Tissues

When dealing with probabilistic models that contain hidden or unobserved variables, the EM algorithm is an iterative approach utilized for parameter estimation. In this context, the term “weights” pertains to the probabilities indicating the likelihood of a given data point belonging to a specific cluster or group. Figure 1 depicts the flow diagram of the EM algorithm.

The steps for computing and updating the weights in the EM algorithm are as follows:

- (1) Initialization: Initialize the parameters of the model, which include the cluster means, covariance, and weights (probabilities).
- (2) E-step: During the E-step of the algorithm, the responsibilities or probabilities that a data point belongs to each cluster are calculated using Baye’s rule. These probabilities are proportional to the product of the prior probability of the cluster and the likelihood of the data point, given the cluster parameters. Subsequently, the responsibilities are normalized to ensure that they add up to 1.
- (3) M-step: In this step, the algorithm updates the cluster parameters, which include the means, covariance, and weights, based on the responsibilities computed in the E-step. The weights are updated by taking the average of the responsibilities for each cluster.

Repeat steps 2 and 3 until convergence: The algorithm iterates through the E-step and M-step until the parameters converge to a stable solution.



**Fig. 1** EM algorithm flow diagram

The EM algorithm is a widely used technique for image segmentation, capable of separating brain tissues. In each iteration of the algorithm, the expectation step calculates the probabilities of each voxel belonging to each tissue type based on the current model parameter estimates. The maximization step updates the model parameters using the expectations computed in the previous step. The algorithm repeats these two steps until either convergence or stopping criteria are met. The mathematical formulation of the EM algorithm is represented in Eqs. (1–4).

$$\mu_p^{\text{new}} = \frac{1}{N_p} \sum_{i=1}^M W_{ip} x^i \quad (1)$$

$$\sigma_p^{\text{new}} = \frac{1}{M_p} \sum_{i=1}^M W_{ip} \cdot (y^i - \mu_p^{\text{new}})(x^i - \mu_p^{\text{new}})'; 1 \leq k \leq K \quad (2)$$

$$k(y|\theta_k) = \frac{1}{(2\pi)^{d/2} |\sum|^{1/2}} e^{-1/2(y-\mu_p)} \sum_{p=1}^P (x - \mu_p); \theta_p : \mu_p, \sum_p \quad (3)$$

$$W_{ip} = \frac{p(y_i|\theta_p) \cdot \alpha_p}{\sum_{m=-1}^P p(y_i|\theta_m) \cdot \alpha_m} \quad (4)$$

In the maximization step, the new mixture weights are computed via the following set of equations, where  $M_p$  is the effective number of observations belonging to the  $p$ th class Eq. (5) and alpha is updated using Eq. (6).

$$M_p = \frac{1}{N} \sum_{i=1}^M W_{i,p} \quad (5)$$

$$\alpha_p = \frac{M_p}{M} \quad (6)$$

Overall, the EM algorithm is used to find the maximum likelihood estimate of the model parameters, which involves maximizing the log-likelihood of the data with respect to the parameters. The weights are updated at each iteration based on the responsibilities, which reflect the probability that each data point belongs to each cluster. The EM algorithm has ability to handle missing data, its flexibility, unsupervised learning capabilities, guaranteed convergence, and ability to estimate uncertainty make it a popular choice for many applications.

## 4 Simulation Results and Discussion

There are various segmentation algorithms available in the literature for extracting regions of interest from CT/MR images of the brain. This research study presents the findings of utilizing the expectation–maximization algorithm to segment MR brain images. The algorithms were developed and simulated using Matlab software (version: 2022). Table 1 represents the performance metrics of the segmentation model.

The algorithm was evaluated on five data sets [31]. The EM algorithm was able to delineate the WM, GM, and CSF from the MR brain images [31, 32]. For performance validation of the algorithm, ground truth images are used. The average value of the dice coefficient was used as a performance metric to validate the segmentation of brain tissues.

Figure 2 depicts the error versus iteration for the segmentation model with respect to the data set D1. Figures 3 and 4 depict segmentation results with respect to the data set D1 (slice numbers 23 and 13). Figures 5 and 6 represent the segmentation results with respect to the data set D2 (slice number 16 and 26).

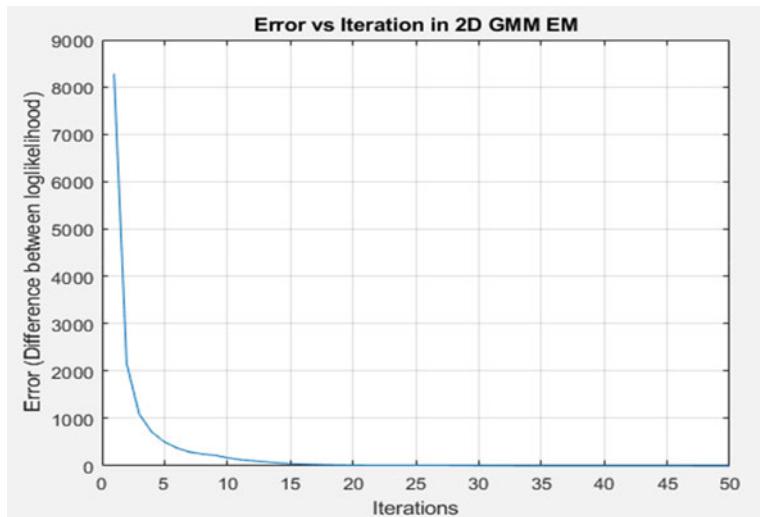
Figures 7 and 8 depict the segmentation results with respect to the data set D3 (slice number 27 and 43).

Figures 9 and 10 depict the segmentation results with respect to the data set D4 (slice number 20 and 32).

The dice coefficient is a measure of percentage of overlap between the ground truth and machine segmented image. Closer the value of DC to 1, better is the segmentation result. Table 1 results reveal that EM algorithm was found to be better in the extraction of CSF than white matter and grey matter. The deep learning-based segmentation model is widely used in many applications [26, 30] for the segmentation and classification of data.

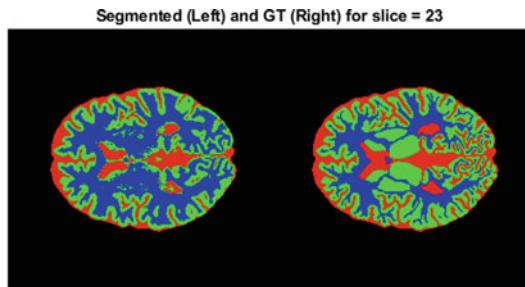
**Table 1** Performance metrics of the segmentation model

ID	Avg. DSC for CSF	Avg. DSC for GM	Avg. DSC for WM	Elapsed time (s)
1	0.79494	0.5562	0.58609	89.328389
2	0.76787	0.48794	0.54119	74.979662
3	0.71133	0.45764	0.53723	81.090592
4	0.72711	0.38342	0.45988	85.380637
5	0.75529	0.41519	0.52224	77.088800



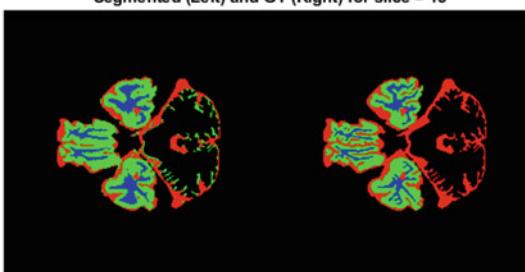
**Fig. 2** Error vs iteration in 2D GMM-EM for the data set D1

**Fig. 3** ROI extraction result of D1 for slice 23



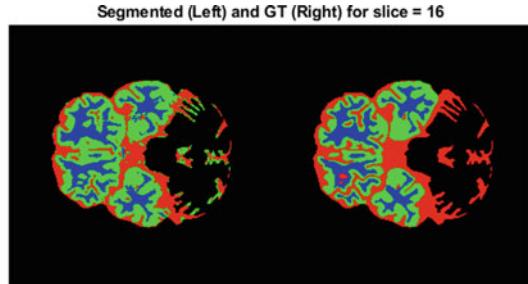
DSC for CSF = 0.79545, DSC for GM = 0.65684, and DSC for WM = 0.70945

**Fig. 4** ROI extraction result of D1 for slice 13



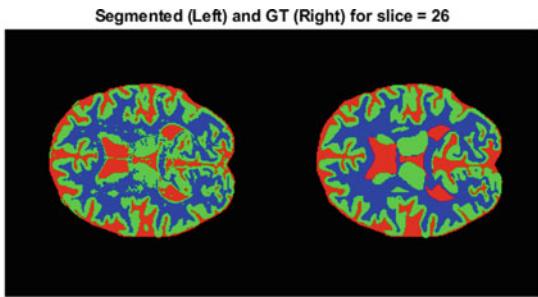
DSC for CSF = 0.7975, DSC for GM = 0.78302, and DSC for WM = 0.77927

**Fig. 5** ROI extraction result of D2 for slice 16



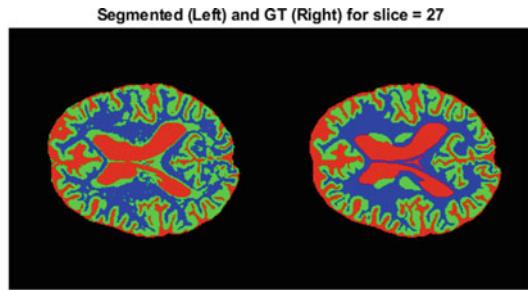
DSC for CSF = 0.79744, DSC for GM = 0.79746, and DSC for WM = 0.86365

**Fig. 6** ROI extraction result of D2 for slice 26



DSC for CSF = 0.86144, DSC for GM = 0.80936, and DSC for WM = 0.8501

**Fig. 7** ROI extraction result of D3 for slice 27

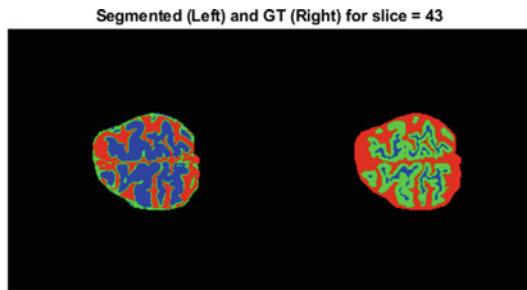


DSC for CSF = 0.88858, DSC for GM = 0.72887, and DSC for WM = 0.76193

#### 4.1 Comparative Analysis of the EM Algorithm with the Deep Learning Model

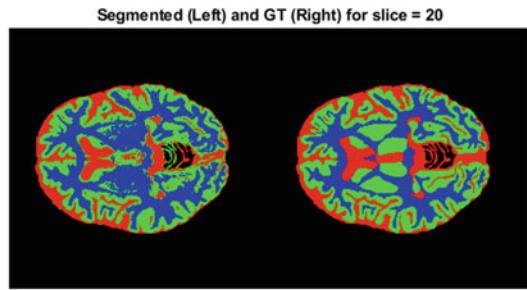
A qualitative comparative analysis of the EM algorithm with the deep learning model is as follows;

**Fig. 8** ROI extraction result of D3 for slice 43



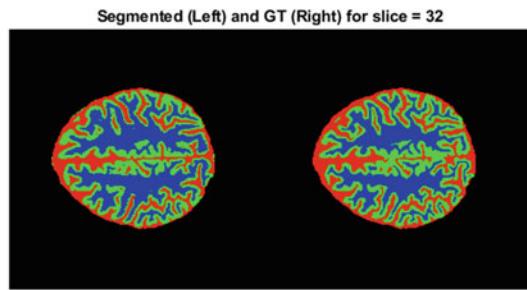
DSC for CSF = 0.81403, DSC for GM = 0.26274, and DSC for WM = 0.30305

**Fig.9** ROI extraction result of D4 for slice 20



DSC for CSF = 0.82862, DSC for GM = 0.74174, and DSC for WM = 0.79497

**Fig.10** ROI extraction result of D4 for slice 32



DSC for CSF = 0.88182, DSC for GM = 0.81725, and DSC for WM = 0.87193

#### 4.1.1 Handling of Missing Data

The EM algorithm can handle missing data or incomplete data by estimating the missing values using the expectation of the latent variables. In contrast, deep learning-based segmentation models require complete data to learn the segmentation task.

#### 4.1.2 Model Complexity

Deep learning-based segmentation models are more complex than the EM algorithm as they use deep neural networks with multiple layers to learn the segmentation task. The EM algorithm, on the other hand, uses probabilistic models to estimate the parameters.

#### 4.1.3 Training Data Requirement

Deep learning-based segmentation models require a large amount of training data to learn the task effectively. In contrast, the EM algorithm can work with small data sets and can still provide good results.

#### 4.1.4 Accuracy

Deep learning-based segmentation models generate proficient results for real-time classification problems with large training data sets. However, EM algorithm can also provide competitive results, particularly for problems with small data sets or missing data.

#### 4.1.5 Interpretability

The EM algorithm provides interpretable results as it estimates the probability of each pixel belonging to each class. In contrast, deep learning-based segmentation models are considered black-box models, and it is difficult to interpret the results.

### 5 Conclusion

This research work is a study focusing on the importance of segmentation models for the extraction of brain tissues from MR images of the brain. The importance of WM, GM, and CSF in the diagnosis of neuro disorders is discussed in this chapter. The widely used segmentation algorithms for brain tissue segmentation are also discussed in this chapter. The expectation–maximization (EM) algorithm results for the MR brain images are also highlighted in this chapter, validated by the dice coefficient. The EM algorithm was found to extract CSF proficiently in the MR brain images. Currently, hybrid algorithms are gaining prominence that relies on artificial intelligence models for automatic segmentation of brain tissues to pave the way toward brain age prediction.

## References

1. Cercignani M et al. (2001) Segmenting brain white matter, gray matter and cerebro-spinal fluid using diffusion tensor-MRI derived indices. *Magnet Resonance Imaging* 19(9):1167–1172
2. Badat N, Savatovsky J, Charbonneau F, Collin A, Lecler A (2017) Multinodular vacuolating and neuronal tumor of the cerebrum. *Neurology* 89(3):304–305
3. Dogra J, Prashar N, Jain S, Sood M (2018) Improved methods for analyzing MRI brain images. *Netw Biol* 8(1):1–1
4. Taki Y, Thyreau B, Kinomura S, Sato K, Goto R, Kawashima R, Fukuda H (2011) Correlations among brain gray matter volumes, age, gender, and hemisphere in healthy individuals. *PloS One* 6(7):e22734
5. Thacker NA, Jackson A (2001) Mathematical segmentation of grey matter, white matter and cerebral spinal fluid from MR image pairs. *Br J Radiol* 74(879):234–242
6. Somasundaram K, Kalavathi P (2014) Brain segmentation in magnetic resonance human head scans using multi-seeded region growing. *The Imaging Sci J* 62(5):273–284
7. Tudorascu DL et al. (2016) Reproducibility and bias in healthy brain segmentation: comparison of two popular neuroimaging platforms. *Front Neurosci* 10. <https://doi.org/10.3389/fnins.2016.00503>
8. Manikandan S, Ramar K, Iruthayarajan MW, Srinivasagan KG (2014) Multilevel thresholding for segmentation of medical brain images using real coded genetic algorithm. *Measurement* 1(47):558–568
9. Cabezas M, Oliver A, Lladó X, Freixenet J, Cuadra MB (2011) A review of atlas-based segmentation for magnetic resonance brain images. *Comput Methods Programs Biomed* 104(3):e158–e177
10. Ibtehaz N, Rahman MS (2020) MultiResUNet: rethinking the U-Net architecture for multi-modal biomedical image segmentation. *Neural Netw* 1(121):74–87
11. Ma G, Yue X (2022) An improved whale optimization algorithm based on multilevel threshold image segmentation using the Otsu method. *Eng Appl Artif Intell* 113:104960
12. Sandhya G, Kande GB, Satya ST (2021) Segmentation of WM, GM, and CSF from the brain MRIs using an advanced metaheuristic approach. *IETE J Res* 10:1–27
13. Ye Z, Song Z, Li P, Wang M, Hou W (2022) A modified threshold score-based multilevel thresholding segmentation technique for brain magnetic resonance images using opposition-based learning hybrid rice optimization algorithm. *Int J Imaging Syst Technol*
14. Jena B, Nailk MK, Panda R, Abraham A (2022) A novel minimum generalized cross entropy-based multilevel segmentation technique for the brain MRI/dermoscopic images. *Comput Biol Med* 151:106214
15. Rai R, Das A, Dhal KG (2022) Nature-inspired optimization algorithms and their significance in multi-thresholding image segmentation: an inclusive review. *Evol Syst* 13(6):889–945
16. Li M, Zhou J, Wang D, Peng P, Yu Y (2022) Application of clustering-based analysis in MRI brain tissue segmentation. *Comput Mathemat Methods Med*
17. Rahman JS, Selvaperumal SK (2023) Integrated approach of brain segmentation using neuro fuzzy k-means. *Indonesian J Electri Eng Comput Sci* 29(1):270–276
18. Tavakoli-Zaniani M, Sedighi-Maman Z, Zarandi MH (2021) Segmentation of white matter, grey matter and cerebrospinal fluid from brain MR images using a modified FCM based on double estimation. *Biomed Signal Process Control* 68:102615
19. Cuadra MB, Pollo C, Bardera A, Cuisenaire O, Villemure JG, Thiran JP (2004) Atlas-based segmentation of pathological MR brain images using a model of lesion growth. *IEEE Trans Med Imaging* 23(10):1301–1314
20. Zhu Q, Wang Y, Du B, Yan P (2022) OASIS: one-pass aligned atlas set for medical image segmentation. *Neurocomputing* 22(470):130–138
21. Bondiau PY, Malandain G, Chanalet S, Marcy PY, Habrand JL, Fauchon F, Paquis P, Courdi A, Commowick O, Rutten I, Ayache N (2005) Atlas-based automatic segmentation of MR images: validation study on the brainstem in radiotherapy context. *Int J Radiation Oncol\* Biol\* Phys* 61(1):289–98

22. van der Lijn F, Den Heijer T, Breteler MM, Niessen WJ (2008) Hippocampus segmentation in MR images using atlas registration, voxel classification, and graph cuts. *Neuroimage* 43(4):708–720
23. Chuang CC, Lee YT, Chen CM, Hsieh YS, Liu TC, Sun CW (2012) Patient-oriented simulation based on Monte Carlo algorithm by using MRI data. *Biomed Eng Online* 11(1):1–6
24. Prakash RM, Kumari RS (2018) Modified expectation maximization method for automatic segmentation of MR brain images. In: Conference: MRBRAINS13, Japan
25. Ramya J, Maheswari BU, Rajakumar MP, Sonia R (2022) Alzheimer's disease segmentation and classification on MRI brain images using enhanced expectation maximization adaptive histogram (EEM-AH) and machine learning. *Inform Technol Control* 51(4):786–800
26. Tuan TA, Bao PT, Kim JY, Tavares JM (2019) White matter, gray matter and cerebrospinal fluid segmentation from brain 3D MRI using B-UNET. In: VipIMAGE 2019: proceedings of the VII ECCOMAS thematic conference on computational vision and medical image processing, October 16–18, 2019, Porto, Portugal 2019, Springer International Publishing, pp 188–195
27. Liu Z, Tong L, Chen L, Jiang Z, Zhou F, Zhang Q, Zhang X, Jin Y, Zhou H (2022) Deep learning based brain tumor segmentation: a survey. *Complex Intell Syst* 9:1–26
28. Nguyen DM, Vu HT, Ung HQ, Nguyen BT (2017) 3D-brain segmentation using deep neural network and Gaussian mixture model. In: 2017 IEEE winter conference on applications of computer vision (WACV) 24 March 2017, IEEE, pp 815–824
29. Kong Z, Luo J, Xu S, Li T (2018) Automatic tissue image segmentation based on image processing and deep learning. In: Neural imaging and sensing 2018, 12 Feb 12 2018, vol 10481. SPIE, pp 79–85
30. Yamanakkanavar N, Lee B (2020) Using a patch-wise m-net convolutional neural network for tissue segmentation in brain MRI images. *IEEE Access* 1(8):120946–120958
31. <https://github.com/kamruleee51/Multi-modal-MRI-Image-Segmentation-EM-algorithm>. 13 Jan 2023
32. Ghosh TK, Hasan MK, Roy S, Alam MA, Hossain E, Ahmad M (2021) Multi-class probabilistic atlas-based whole heart segmentation method in cardiac CT and MRI. *IEEE Access* 3(9):66948–66964

# Detection of Cyberattacks in Cyber-Physical Systems Using Supervised Learning and Hypergraphs



S. Priyanga, S. Pravinraj, Venkata Bhavana Repalle, Kannan Krishivasan,  
and V. S. Shankar Sriram

**Abstract** Critical Infrastructures (CIs) include Supervisory Control and Data Acquisition (SCADA) systems that gather and monitor real-time data. Recent advancements in CIs enable two-way dialog, automated monitoring, and easier energy supply management. However, without robust security approaches, smart grids are more vulnerable to cyber-threats and cyberattacks. In this context, the researchers focus on developing machine learning approaches for early attack detection with better detection rate and reduced computational overhead. However, most of the security approaches are not helpful in identifying all the SCADA-specific attack vectors. Furthermore, most of the research works focus on the performance, stability, robustness, and efficiency of physical systems rather than security, which is frequently overlooked due to constrained factors such as limited processing and storage capacity. Hence, this work presents Hyperclique–Binary Salp Swarm Algorithm-based Support Vector Machine (HC-BSSA SVM), an adaptive attack detection model for cyber-attack detection in Industrial Control System (ICS) with a high detection rate and less false positives.

**Keywords** Critical Infrastructures · Cyber-Physical Systems · Industrial Control Systems · Attack detection · Feature selection

---

S. Priyanga

Department of Computer Science and Engineering, Srinivasa Ramanujan Centre, SASTRA  
Deemed University, Kumbakonam 612001, India  
e-mail: [priyanga@src.sastra.edu](mailto:priyanga@src.sastra.edu)

S. Pravinraj

Incedo Inc., Chennai 600096, India

V. B. Repalle

College of Business, The University of Tampa, Tampa 33592, USA

K. Krishivasan · V. S. Shankar Sriram (✉)

School of Computing, SASTRA Deemed University, Thanjavur 613401, India  
e-mail: [sriram@it.sastra.edu](mailto:sriram@it.sastra.edu)

## 1 Introduction

Cyber-Physical System (CPS) is an electronic control system that integrates the cyber and physical components that can effectively modernize CIs using sensors, networks, and computing technologies [1]. The security of CPS differs from IT security as it prioritizes availability rather than confidentiality. However, the misclassification of normal and attack operation can limit the availability of ICS. This principle of CPS opens room for vulnerabilities and targets the CIs, leading to economic losses [2].

According to ICS CERT, “The number of cyberattacks against ICS systems has significantly increased over the past few years.” Safeguarding the CIs from the exposed threats is not easy, and the increase in cyberattacks against CI has shown that ICS became a hotspot for attackers to inject new threats and vulnerabilities. Understanding the criticality and severity of the cyberattacks against ICS is significant for the earlier evaluation of attack detection algorithms. The considerable research has been conducted to produce promising solutions for the earlier attack detection.

Considering the limitations in existing machine learning techniques, this article presents a novel attack detection approach to ensure protection in CIs with a better detection rate and less false alarm rate. The significant contributions of the proposed approach are listed as follows:

1. This research work presents an efficient cyberattack detection approach for SCADA power systems using Hyperclique–Binary Salp Swarm Algorithm and Support Vector Machine.
2. The proposed model learns historical data to detect the normal and abnormal behaviors in power systems.
3. Hyperclique, a hypergraph variant is introduced to identify the tractable features toward malicious data that finds the optimal features with minimal time.
4. The high-dimensional nature of the cyber-physical systems degrades the performance of the detection mechanisms. Hence, SVM classifier is designed by optimizing its kernel functions for detecting cyberattacks in power grids. Kernel parameter optimization in SVM helps in achieving better classification accuracy and less false alarm rate.
5. In particular, the proposed approach combines feature selection and parameter optimization techniques for detecting cyberattacks and to achieve the global optimal solution with faster convergence.
6. The performance of HC-BSSA SVM has been validated using Mississippi’s power system dataset in terms of classification accuracy, precision, recall, and F-score. The proposed model provides average accuracy of 95.87% on 15 sets.

Organization of the paper: Sect. 2 discusses the existing attack detection approaches present in literature. Section 3 explains the work behind the HC-BSSA SVM. Experimental analysis is provided in Sect. 4. Section 5 discusses the results and discussions of HC-BSSA SVM, and Sect. 6 concludes the paper.

## 2 Related Works

Gumaei et al. [3] proposed correlation feature selection (CFS) method for dimensionality reduction and KNN instance-based learning (IBL) algorithm to classify normal and attack events. The proposed method has reduced dimensionality in the power system dataset, and accuracy is maximized with the KNN learning model. However, the computational cost is very high. The temporal patterns discovery algorithm is proposed by Kalech et al. [4] to identify malicious activities through frequent temporal patterns mining and state-based automaton. The proposed algorithm can detect the attacks which change the register's values. During the mining process, the exponential growth of patterns is discovered, which leads to high computational overhead, and the discretization approach causes data loss. Graph network (GN)-based detection mechanism is developed by Li et al. [5] to classify normal and tampered measurements which helps to identify the location of False Data Injection (FDI) attacks. The performance of Caps-GN is better in determining the location of the attack with an increased attack scale. Though the stability and accuracy of the proposed approach are better, the false alarm rate is high when the attack meter is more than 15. Chen et al. [6] designed an imperfect data injection strategy to construct an unidentifiable attack with minimum residual increment. The PSO algorithm is introduced to reduce the computational overhead, and a new chi-square detection approach with two state estimators is proposed to detect false data injections. The proposed method detects attacks effectively, even with noise inferences, and eliminates the false alarms incurred due to noises. However, it is unable to solve unknown attack vectors. The work of Ashrafuzzaman et al. [7] adapts the ensemble machine learning approach, which uses both supervised and unsupervised classifiers to identify stealthy data injection attacks in smart grids. It has resolved the “curse of dimensionality” issue which increases the training speed and detects the attacks more quickly. The authors considered the advantages of both supervised and unsupervised classifiers. Based on the sensitivity score, the proposed scheme identifies only 73% of attacks. Mostafa et al. [8] brings Principal Component Analysis (PCA) for dimensionality reduction and k-NN for binary classification which suffers from high false positives when system configuration changes dynamically. Raman et al. [9, 10] have proposed two approaches for anomaly detection in water treatment plants. Deep auto-encoder-based approach produces fewer false positives and high detection rate. However, Cumulative Sum (CUSUM)-based approach which is integrated with multilayer perceptron helps in identifying deviations in sensor values. The proposed approach is able to detect stealthy data injections, and also false alarm rate is less. However, it is limited to identify single point attacks and not for multipoint attacks. Jagtap et al. [11] proposed parallel Kohonen map neural network to identify zero-day attacks with less computational overhead and it failed to generalize imbalanced data.

From the literature study, we can observe that classification accuracy is the center of attention to the researchers as it remains a significant concern in data-driven techniques proposed in CPS security. Therefore, the primary motivation of this work is

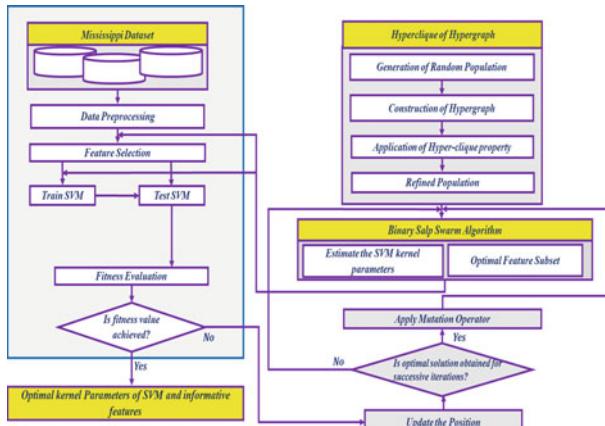
to develop a novel attack detection approach to overcome the problems that traditional statistical and machine learning approaches [12] cannot identify. Unsupervised learning has advantages on identifying anomalies. It is more susceptible to false alarms, and supervised learning approaches play a promising role in attack detection with fewer false positives [13]. Among those approaches, SVM is the prominent choice for researchers as it adapts the “structural risk minimization” principle and is less prone to overfitting issues [14]. Hence, this paper has taken advantage of supervised learning and proposed a novel attack detection approach where the parameters of SVM is optimized with a binary salp swarm algorithm.

### 3 HC-BSSA SVM: The Proposed Signature-Based Attack Detection Approach

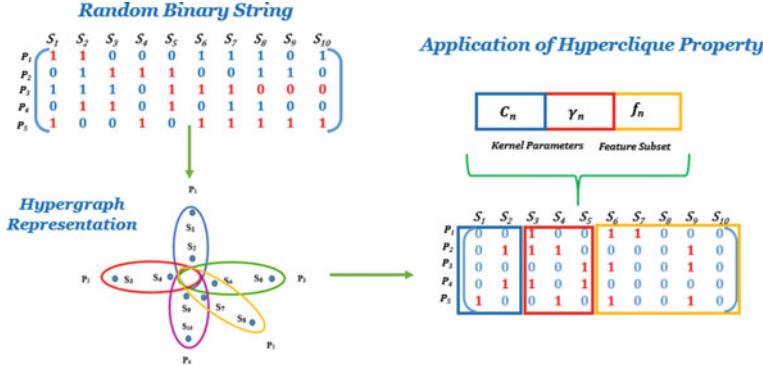
The proposed framework aims to identify the informative features which helps in identifying attack vectors. It involves two stages: (i) feature selection and parameter optimization and (ii) classification of attacks (Fig. 1).

The major objective of feature selection is to identify informative features from the original feature set. Exploiting hyperclique [11] property on the constructed hypergraph generates the optimal ‘1’s when the population is initialized (Fig. 2). The use of hyperclique property helps to achieve the optimal fitness value which further minimizes the convergence time of BSSA.

**Step 1. Initial Population Generation:** A random two-dimensional matrix of order  $G^* H$  is constructed where ‘ $G$ ’ denotes the maximum population size and each population size is denoted by ‘ $H$ ’. Each bit in the population is represented as vertices of the hypergraph, and it is constructed by bringing the neighborhood



**Fig. 1** Working model of the proposed HC-BSSA SVM



**Fig. 2** Hypergraph representation of feature selection and parameter optimization

relations between  $G$  and  $H$ . Hyperclique property of hypergraph is applied to initialize each population with optimal ‘1’s where ‘1’ represents the presence of features.

**Step 2. Kernel parameter optimization:** In HC-BSSA SVM, each salp in the population was represented as a binary vector, and it breaks up into three major parts:  $[M_C^1, M_C^{C_n}]$ ,  $[M_\gamma^1, M_\gamma^{\gamma_n}]$ , and  $[M_{f_s}^1, M_{f_s}^{f_n}]$  where  $C$  and  $\gamma$  represent the kernel parameters of SVM and  $f_s$  indicates the feature subset, respectively.

**Step 3. Binary transformation:** After the initial population has been generated, the binary vectors  $[M_C^1, M_C^{C_n}]$ ,  $[M_\gamma^1, M_\gamma^{\gamma_n}]$ , and  $[M_{f_s}^1, M_{f_s}^{f_n}]$  are converted into equivalent floating-point values using Eq. (1)

$$\text{Float} = \text{ker}_{\min} + \frac{\text{ker}_{\max} - \text{ker}_{\min}}{2^{CL} - 1} * \text{decimal} \quad (1)$$

$\text{ker}_{\min}$  represents lower bound, and  $\text{ker}_{\max}$  represents the upper bound of kernel parameters.  $CL$  is the code length, and decimal indicates the decimal value of the binary vector. In the case of binary transformation, the values of search space cannot be the decimal equivalent as the values of search positions increase exponentially with an increase in the number of attributes. To overcome this issue, the arctan transfer function is employed to update the position of a salp, where this transfer function is used to update the position of the salp between 0 and 1. To map the continuous features into binary, an arctan transfer is adapted and it is represented in Eq. (2).

$$T(x_i^j) = \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x_i^j\right) \right| \quad (2)$$

The steps involved in classifying the normal samples from attack samples are listed as below:

**Step 1. Training and Testing:** From the original power system dataset  $[D_{M* n}]$ , the training  $[\text{Train}_{TRS*n}]$  and testing  $[\text{Test}_{TES*n}]$  datasets are generated in 80:20 ratio.

**Step 2. Deriving Fitness Function:** The major objective of the proposed HC-BSSA-based SVM is to obtain the higher Detection Rate (DR) and less False Alarm Rate (FAR) with an optimal feature subset. To attain the desired goal, an objective fitness function (Eq. 3) was designed which involves the major parameters, i.e., DR, FAR, and obtained informative attributes that make a great impact on the complexity of the learning model (i.e., maximum weight is assigned to detection rate and minimum weight is assigned to false alarm rate).

$$\text{Fitness} = \left[ W_1 * DR + W_2 * FAR + W_3 \left( 1 - \frac{\sum_{i=1}^N f_i}{N} \right) \right] \quad (3)$$

where

' $N$ ' represents the number of features;  $f_i$  indicates the presence (1's) and absence (0's) of  $i$ th attribute, and  $W_1$ ,  $W_2$  and  $W_3$  represent the predefined weights of DR, FAR, and informative features.

**Step 3. Termination Condition:** The proposed approach returns the optimal kernel parameters of SVM and the informative features when the algorithm reaches the maximum fitness value or the number of generations.

**Step 4. Position update:** The position of each salp was generated based on the binary search space. ('0' or '1'). The position update of BSSA is similar to SSA [15]. Hence, the position is updated using Eq. (4)

$$B(x_i^j) = \begin{cases} 1, & \text{rand} < \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2} x_i^j\right) \right| \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$B(x_i^j)$  denotes the binary value at  $j^{th}$  dimension.

#### Algorithm 1: Proposed HC-BSSA SVM

HC-BSSA SVM()

1. Begin
2. Generate Training ( $\text{Train}_{TRS*n}$ ) and Testing dataset ( $\text{Test}_{TES*n}$ );
3. Initialize the population of the salp ( $i = 1, 2, \dots, n$ ); consider upper bound = 1 and lower bound = 0;
4. Initialize fitness value  $\leftarrow 0$ , maximum number of generations, and assign weights to DR, FAR, and optimal feature subset.
5. Construct the hypergraph and apply hyperclique property on Hypergraph
6. **While** ( $t \leq N_{\text{iter}}$ ) Begin
7. **for each**  $i = 1$  to  $N_{\text{Pop size}}$
8. **for each** binary salp  $S(j)$ : Obtain decimal equivalent using Eq. (1)
9. Calculate V-shape binary transfer using Eq. (2)

10. If random < V-shaped transfer function:

Update  $S_{(i,j)} = 1$ .

Else  $S_{(i,j)} = 0$ .

**End**

**End for**

11. Compute fitness value using Eq. (3)

12. Update  $r_1$  by  $r_1 = 2e^{-(4n/N)^2}$

13. **for each** search agent ( $M_i$ )

14. **if (i == 1)**

Mutate the position of the leader salp with probability  $p_a$  by

$$M_i^j = \frac{1}{2}at^2 + V_0t$$

Else Update position of the follower salp using  $M_i^j =$

$$\frac{1}{2}(M_i^j + M_j^{i-1})M_i^j = \frac{1}{2}(M_i^j + M_j^{i-1})$$

**End if**

If  $fitness(M_i) > fitness(F)$ .

**then**  $F = M_i$

**End for**

**End while**

15. Return the optimal  $(C, \gamma)$ . and informative feature subset

**End**

However, to improve the exploration and exploitation of the search space, we bring out a conflict-based mutation operator. When the condition is true, the position of the leader salp has been mutated with probability  $p_a$  by Eq. (5).

$$M_i^{t+1} = \text{Mutation}(M_i^{t+1}) \quad (5)$$

As a whole, using hyperclique property helps to improve the hyper-relations among the attributes and samples in less number of hyperedges. Hence, it identifies the informative feature vector that detects malicious data in less time with better accuracy.

## 4 Experimental Analysis

The experiments were carried out using an Intel ® Core™ i3-6100U CPU @ 2.30 GHz processor running on 64-bit windows 10 operating system with 8 GB RAM and implemented in Python 3.5. Further, the Weka tool is used for validating few of the existing approaches. The experiments of the proposed method are carried out using the power system attack dataset and gas pipeline dataset [16].

### 4.1 Power System Attack Dataset

This dataset is simulated by the Mississippi State University and Oak Ridge National Laboratory which contains three classes of data that are derived from one initial raw dataset where it includes 15 sets with 37 power system event scenarios and 128 features for each set. The 37 event scenarios are categorized into (i) natural events scenario (8 events), (ii) no events scenario (1 event), and (iii) attack events scenario (28 events). Four types of attack event scenarios are considered: (i) short-circuit fault, (ii) line maintenance, (iii) remote tripping command injection, (iv) relay setting change, and (v) data injection. One percent of the data is randomly sampled and clustered into binary classes which we used in our experiments. In binary class, natural and no events scenario are considered as normal class and labeled as ‘0’ and attack events are labeled as ‘1.’

### 4.2 Gas Pipeline Dataset

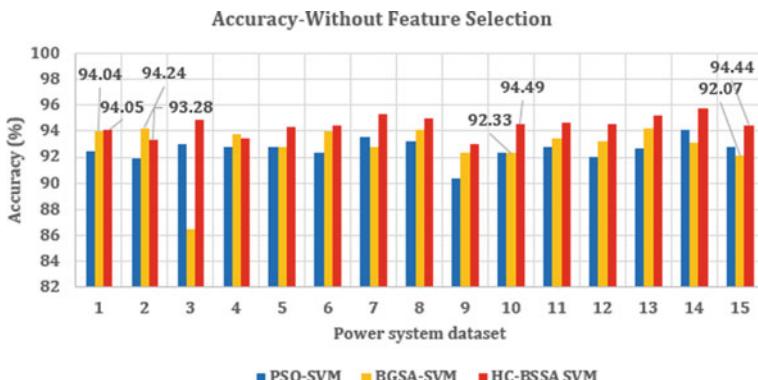
The gas pipeline dataset is collected from the Mississippi State University’s in-house SCADA laboratory. A total of 2,74,628 records have been collected under two modes of operation. In normal mode, 2,14,580 records were collected, and Response injection, command injection, denial of service (DoS), and reconnaissance attacks were injected to collect 60,048 samples under attack operation.

## 5 Results and Discussions

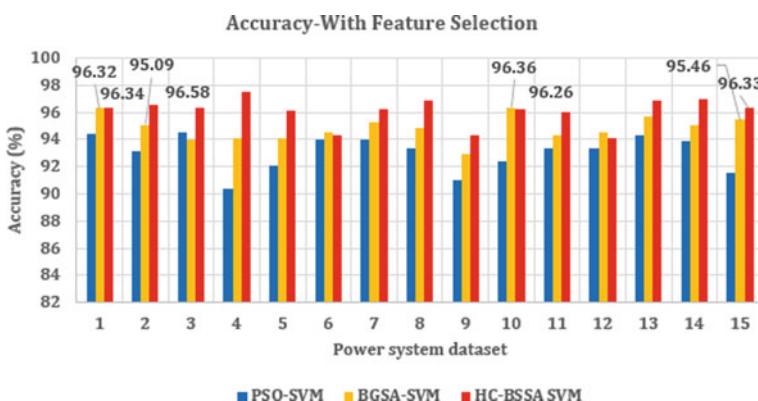
In general, the effectiveness of an attack detection approach is assessed by its detection accuracy and false positives. Therefore, the experiments were implemented under two cases: (i) SVM trained with all attributes and (ii) SVM trained with relevant attributes gained from HC-BSSA. The first experiment was carried out with the optimal kernel parameters of SVM and 128 features in the original dataset. The second experiment was carried out with the optimal kernel parameters of SVM and the

informative features obtained from feature selection. The performance of HC-BSSA SVM has been compared with existing approaches (i) Particle Swarm Optimization-based SVM (PSO-SVM), (ii) Binary Gravitational Search-based SVM (BGSA-SVM), (iii) J48, and (iv) MLPNN. The first two approaches perform parameter optimization and also feature selection.

Figures 3 and 4 present the classification accuracy for both scenarios by HC-BSSA SVM. We can observe that the performance of HC-BSSA SVM is comparatively dominant to the other existing approaches, and for a few sets, the existing BGSA-SVM performs better than the HC-BSSA SVM, e.g., 96.32% accuracy is obtained by BGSA-SVM, whereas HC-BSSA SVM achieved 95.34% for set 1; in set 6, BGSA-SVM: 94.53%, and HC-BSSA SVM: 94.31%, and in set 15, HC-BSSA with 94.33% which is lesser than 95.46% of BGSA-SVM. Though the classification results without feature selection are almost similar between BGSA-SVM and HC-BSSA SVM, the proposed approach shows its efficiency in terms of detection and false alarm rates.



**Fig. 3** Classification accuracy—without feature selection

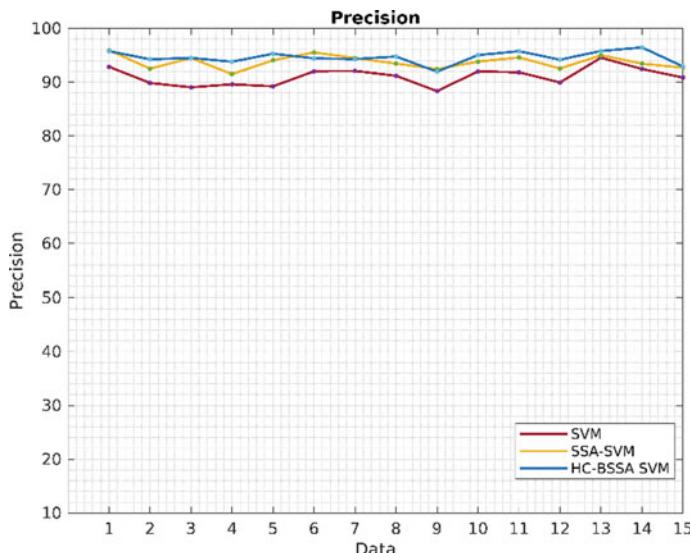


**Fig. 4** Classification accuracy—with feature selection

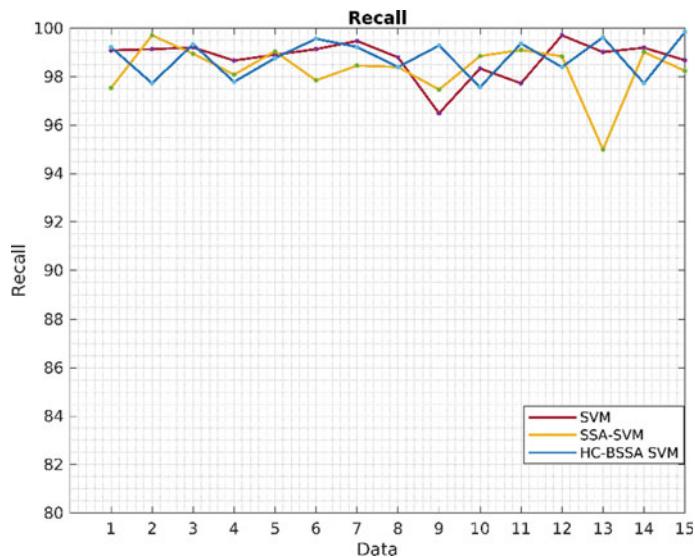
### 5.1 Impact of Hyperclique Property of Hypergraph

Hyperclique property plays a vital role in tuning the kernel parameters and a large number of attributes is reduced to optimal feature subset. Table 3 presents the performance evaluation of HC-BSSA SVM. From this table, we can observe that binary search space reduced the epochs of convergence with optimal gamma factor falling in the range of 2–8. The penalty factor and feature subset selected for each dataset converged variably according to the types of attack data present in the temporal distribution. Also, the feature selection helps in reducing training and testing time as well. The performance of the proposed HC-BSSA SVM is compared with SVM and SSA-SVM in terms of precision, recall, and F-score to understand the impact of mutation operator and hyperclique property in hypergraph (Figs. 5, 6, and 7). An average F-score obtained by HC-BSSA SVM is 96.65% which is higher than the values achieved by SVM is 94.74% and SSA-SVM is 96.08%. Similarly, recall value obtained by the proposed approach is 98.79% which determines the detection efficiency. The proposed HC-BSSA SVM is able to identify the attack class with an average of 18.83 secs. The major intuition behind using hyperclique property is to reduce the computational overhead, and it is evident from the observations that the desired goal is achieved.

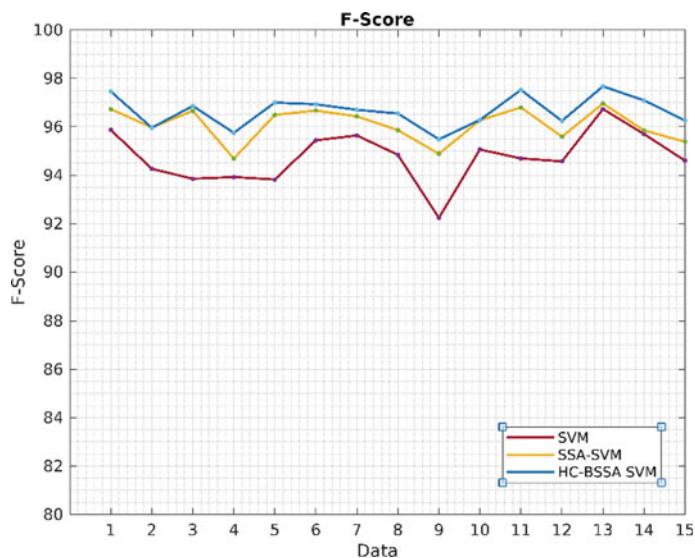
To ensure the adaptability of the proposed attack detection approach for various ICS, Mississippi's gas pipeline dataset is also considered for experimental evaluation (Table 1). To determine the statistical significance of the proposed HC-BSSA SVM, the Kruskal–Wallis significant test [17] is examined and the results are compared with the existing SVM and SSA-SVM methods.



**Fig. 5** Precision



**Fig. 6** Recall



**Fig. 7** F-score

**Table 1** Performance evaluation for gas pipeline dataset

Classification techniques	Accuracy (%)	Precision	Recall	False alarm rate
PSO-SVM	95.55	95.20	97.82	0.08
BGSA-SVM	95.17	94.61	97.81	0.09
J48	87.17	85.23	97.32	0.31
MLPNN	91.23	90.46	97.52	0.38
HC-BSSA SVM	<b>98.32</b>	<b>97.14</b>	<b>98.88</b>	<b>0.05</b>

Bold indicates the significance of the proposed approach

**Table 2** Performance comparison with existing approaches

Authors (%)	Accuracy	Precision	Recall	F-Score
Wang et al. [18]	90.41	93.8	93.6	93.5
Eirni anthi et al. [19]	94.14***	94.0***	94.0***	94.0***
Manikant Panthi et al. [20]	94.36**	92.1**	98.9*	95.4**
Proposed HC-BSSA SVM	<b>95.87*</b>	<b>94.6*</b>	<b>98.79**</b>	<b>96.65*</b>

Bold indicates the significance of the proposed approach

\* indicates the ranking of the attack detection approaches. \* indicates first position; \*\* indicates second and \*\*\* indicates third position

The difference between the mean ranks of HC-BSSA SVM and SSA-SVM is 10.23 and for SVM and HC-BSSA SVM is 25.87 which is large. The test score is close to zero, showing no dependency among the samples. Also, it proves HC-BSSA SVM outperforms both SVM and SSA-SVM. Also, Kruskal–Walli's value is greater than the critical chi-square value which rejects the null hypothesis by 0.0009 significance score and accepts the alternative hypothesis. From this, it is evident that the accuracy is different and statistically significant among these approaches. A total of 94.60% precision rate is obtained by the proposed approach, which is 3% higher than the existing approaches. For the gas pipeline dataset, 98.32% accuracy is obtained for the proposed HC-BSSA SVM, which is 3% higher than BGSA-SVM and PSO-SVM, and 98.88% recall value was obtained, which is 2% higher than BGSA-SVM and PSO-SVM.

The performance of the proposed approach is compared with the recent attack detection approaches (Table 2.). HC-BSSA SVM has achieved 95.87% of accuracy which is higher than the other three existing approaches. Though, the recall value of [20] is ranked top with 98.9%, HC-BSSA SVM holds its dominance in terms of F-score (Table 3).

**Table 3** Performance evaluation of HC-BSSA SVM

Dataset	HC-BSSA SVM parameters				No. of selected features	False alarm rate
	Initial $C$	Initial $\gamma$	Optimal $C$	Optimal $\gamma$		
Set 1	22.0	4.0	60.0	4.0	79	0.007
Set 2	30.0	6.0	12.0	2.0	54	0.022
Set 3	46.0	10.0	36.0	2.0	77	0.006
Set 4	56.0	10.0	58.0	2.0	54	0.022
Set 5	40.0	14.0	58.0	6.0	48	0.012
Set 6	62.0	14.0	52.0	2.0	79	0.004
Set 7	42.0	4.0	44.0	4.0	73	0.007
Set 8	24.0	14.0	8.0	4.0	61	0.015
Set 9	20.0	14.0	20.0	6.0	58	0.007
Set 10	4.0	14.0	16.0	2.0	55	0.024
Set 11	26.0	14.0	36.0	6.0	52	0.006
Set 12	12.0	6.0	8.0	4.0	69	0.015
Set 13	28.0	4.0	60.0	8.0	59	0.003
Set 14	42.0	8.0	36.0	2.0	56	0.022
Set 15	30.0	12.0	52.0	6.0	79	0.001

## 6 Conclusion

This work presented a novel attack detection approach using HC-BSSA SVM which carries out feature selection for power system data and optimizes the kernel parameters in SVM. The proposed HC-BSSA SVM has identified optimal features which improve the efficiency of the classifier. Also, the highest detection rate is achieved with reduced features from 128 smart grid network features, and the false alarm rate is controlled by parameter optimization of SVM. The performance of BSSA has been improved when the hyperclique property of hypergraph is applied to generate the initial population of BSSA and also when the mutation operator is introduced to update the position of the salp. From the experimental results, it is evident that the proposed machine learning model helps to identify attacks in power grids and it was computationally effective. Despite these outcomes, future work is required to deploy the machine learning models in the operational environment.

**Acknowledgment** This work is supported by the Department of Science and Technology ICPS division. (Ref. No: T-615).

## References

1. Ashibani Y, Mahmoud QH (2017) Cyber physical systems security : analysis, challenges and solutions. *Comput Secur* 68:81–97
2. Miller T, Staves A, Maesschalck S, Sturdee M, Green B (2021) Looking back to look forward: lessons learnt from cyber-attacks on industrial control systems. *Int J Critical Infrastruct Protect* 35:100464
3. Gumeai A et al. (2020) A robust cyberattack detection approach using optimal features of SCADA power systems in smart grids. *Appl Soft Comput* 96:106658
4. Shlomo A, Kallech M, Moskovitch R (2021) Temporal pattern-based malicious activity detection in SCADA systems. *Comput Secur* 102:102153
5. Li Y, Wang Y (2020) Developing graphical detection techniques for maintaining state estimation integrity against false data injection attack in integrated electric cyber-physical system. *J Syst Archit* 105:101705
6. Chen R, Li X, Zhong H, Fei M (2019) A novel online detection method of data injection attack against dynamic state estimation in smart grid. *Neurocomputing* 344:73–81
7. Ashrafuzzaman M, Das S, Chakhchoukh Y, Shiva S, Sheldon FT (2020) Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning. *Comput Secur* 97:101994
8. Mohammadpourfard M, Weng Y, Pechenizkiy M, Tajdinian M, Mohammadi-Ivatloo B (2020) Ensuring cybersecurity of smart grid against data integrity attacks under concept drift. *Int J Electr Power Energy Syst* 119:105947
9. Gauthama Raman MR, Dong W, Mathur A (2020) Deep autoencoders as anomaly detectors: method and case study in a distributed water treatment plant. *Comput Secur* 99:102055
10. Raman MRG, Somu N, Mathur AP (2020) A multilayer perceptron model for anomaly detection in water treatment plants. *Int J Crit Infrastruct Prot* 31:100393
11. Jagtap SS, Shankar Sriram VS, Subramaniyaswamy V (2021) A hypergraph based Kohonen map for detecting intrusions over cyber-physical systems traffic. *Futur Gener Comput Syst* 119:84–109
12. Priyanga S, Krishivasan K, Pravinraj S, Shankar Sriram VS (2020) Detection of cyberattacks in industrial control systems using enhanced principal component analysis and hypergraph-based convolution neural network (EPCA-HG-CNN). *IEEE Trans Ind Appl* 56(4):4394–4404
13. Priyanga S, Gauthama Raman MR, Jagtap SS, Aswin N, Krishivasan K, Shankar Sriram VS (2019) An improved rough set theory based feature selection approach for intrusion detection in SCADA systems. *J Intell Fuzzy Syst* 36(5):3993–4003
14. Wang H, Gu J, Wang S (2017) An effective intrusion detection framework based on SVM with feature augmentation. *Knowledge-Based Syst* 136:130–139
15. Berge C (1973) Graphs and hypergraphs. North-Holland Publishing Company, 1st edn. ISBN series: 072042450X, ISBN volume: 0720424534
16. Mirjalili S, Gandomi AH, Zahra S, Saremi S (2017) Salp swarm algorithm : a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
17. Mississippi State University Critical Infrastructure Protection Center (2014) Industrial control system cyber attack data set. Online: [http://www.ece.msstate.edu/wiki/index.php/ICS\\_Attack\\_Dataset](http://www.ece.msstate.edu/wiki/index.php/ICS_Attack_Dataset). Apr 2014
18. McKight J, Najab PE (2011) Kruskal-Wallis test. *Encycl Dict Polym* 1:984–985
19. Anthi E, Williams L, Rhode M, Burnap P, Wedgbury A (2021) Adversarial attacks on machine learning cybersecurity defences in industrial control systems. *J Inf Secur Appl* 58:102717
20. Panthi M (2020) Anomaly detection in smart grids using machine learning techniques. In: 21st International conference power, control computing technology, ICPC2T 2020, pp 220–222

# Classifying Fetal Health Using Neural Networks by Boosting Imbalanced Classes



Perumalla Anoosha, Renuka Devi Parlapalli, E. Srikanth Reddy, and P. Menaga

**Abstract** In recent days, fetal health care has become more precious to giving birth to a child. To maintain the good health of the fetus, the mother needs proper observation and treatment. This process requires the technology assistant to observe the fetus's health continuously. In this paper, we proposed ANN-based fetal health classification model to provide continuous observation of fetal health environmental situations. The dataset related to fetal health was collected from the Kaggle website, which has 2126 samples, along with that synthetic data created with the synthetic minority oversampling technique (SMOTE) method to balance the actual data. It produced 2839 sample dataset records. From that, the model has avoided the dominations of majority classes. This research trained the ANN model of three layers with two hidden layers and compared the accuracy, precision and recall, and F1-score. The accuracy of this model achieved the 92.0%.

**Keywords** Fetal health prediction · SMOTE · ANN · Deep learning · Machine learning

---

P. Anoosha (✉)

School of Computer Science and Artificial Intelligence, SR University, Warangal,  
Telangana 506371, India  
e-mail: [perumalla.anoosha@gmail.com](mailto:perumalla.anoosha@gmail.com)

R. D. Parlapalli

Mother Theressa College of Engineering and Technology, Peddapally, Telangana 505174, India

E. Srikanth Reddy

Vaageswari College of Engineering, Karimnagar, Telangana 505481, India

P. Menaga

Department of Computer Science and Business Systems, Rajalakshmi Engineering College, Tamil Nadu, Chennai 600010, India  
e-mail: [menaga.p@rajalakshmi.edu.in](mailto:menaga.p@rajalakshmi.edu.in)

## 1 Introduction

In the present days, women are facing the problem with fetal health complications that lead to unhealthy fetal growth. This is the major challenging issue in the current scenario of health care. There are many pregnancies related deaths and abortions that have been recorded in the world, this did not occur only in a specific area but in a large, dispersed area [1]. Usually these cases used to occur only in the areas where the people did not have access to good medical facilities or good monitoring. These cases are now not only occurring in destitute women but also in affluent women. Thus, among the main causes of death in poorer nations are complications related to pregnancy and delivery [2, 3].

While some of these issues may arise prior to conception but worsen during pregnancy, most of these issues arise during pregnancy. Pregnancy complications might include high blood pressure issues, gestational diabetes, infections, hypertension, miscarriage and pregnancy loss, premature labor, and stillbirth. Severe nausea, vomiting, and iron deficiency anemia are further concerns [3–5]. As a result, these disorders may have an impact on pregnancy, necessitating the need for new methods of screening and evaluating fetal well-being [6]. However, the majority of these maternal fatalities during pregnancy might have been avoided or treated because they virtually all happened in low-resource environments.

There are many ways to monitor the fetal growth and its health condition using the resources available in today's time. Cardiotocography (CTG) is a technological way of continually detecting and recording the fetal heart rate (FHR) and uterine contractions throughout pregnancy to assess fetal well-being and monitor for increased risk of pregnancy problems [6, 7]. This technology provides monitoring the progression of fetal hypoxia and responding effectively before severe asphyxia or death occurs. There have been a lot of study publications published in the literature on autonomous fetal movement recording utilizing various types of sensors put on the maternal abdomen, such as accelerometers, acoustic sensors, and so on. This is because fetal movement can induce sound waves or oscillations on the mother abdominal wall [8–10]. Furthermore, they offer advantages such as being passive and non-invasive (in comparison with ultrasound images), cheap cost, and light weight, making them a suitable choice for fetal movement signal gathering. Recently, sophisticated technologies in current medical practices have effectively enabled the use of robust and accurate machine learning and artificial intelligence approaches in delivering automated prediction based on early detection results in a variety of medical applications [11, 12].

Various machine learning techniques have been applied to observe the pattern and abnormalities in fetal growth by using various datasets available throughout the repositories by some of the researchers, but many of the researchers have preferred to make their own dataset by considering the samples collected in real time by recording the data of pregnant women by their consent [13–15]. These data is used to train the machine learning models that are selected by the researchers, the number of models may depend on the criteria of each individual. Majority of simple machine learning

models were used on the data where the data was in numeric format and the data where it is in image or audio format some pre-processing techniques and feature extraction techniques were used to convert the data into numerical format. Many authors and researchers combined the high dimensional data to get the maximum accuracy possible [16].

The models have widely used decision tree, random forest, neural networks, etc. The proposed methodology of each individual usually consisted of similar steps: data acquisition, feature identification, model training, and model evaluation. Some authors also preferred integrating the models to get better accuracy combined. Therefore, machine learning can be used to improve the current scenario in the healthcare sector, specifically in determining the fetus's condition and recommending women for regular health checkups.

In our approach, we have introduced a model to classify the fetal health assistant model to assist the mother to get proper guidance in child growth [17]. We have collected the initial dataset from Kaggle then applied the SMOTE method from that we have created a synthetic dataset of 4000+ samples records of numerical data. We have created synthetic data with SMOTE method to balance the actual data, and it produced 4000+ records. We trained the ANN model to classify the data and achieved the better results. Rest of the paper has discussed about literature review in Sect. 2. Dataset classification and methodology has explained in Sect. 3, and finally, Sect. 4 has included with results and analysis.

## 2 Literature Review

Akbulut et al. [1] have aimed to provide assistive services to pregnant women and clinicians via an online predictive system which helps to predict the fetal health. The dataset has been collected on their own and no previously available dataset has been used in this work, the dataset they collected consisted data of 96 pregnant women which has 22 input features and 1 output label which determines the health of the fetus. Multiple classification algorithms have been used and compared, and the best accuracy model has been chosen by the authors to be deployed. The classification algorithm gives an accuracy of 89.5% about the fetal health status using the two-class Decision Forest algorithm. In real-time testing with 16 users, the performance was 87.5%.

Miao et al. [2] improved artificial intelligence technique based on multiclass morphologic pattern predictions and deep learning classification models is suggested for cardiotocographic diagnosis and prenatal evaluation. The created model is utilized to differentiate and categorize the presence or absence of multiclass morphologic patterns for predicting pregnancy problems. The dataset for this work was obtained from the CTG databases, which can be found at the UCI machine learning repository. In the CTG dataset, there are a total of 2126 clinical occurrences indicating various pregnancy problems on fetal cardio. The CTG dataset comprises 21 input attributes, one multiclass attribute, and one fetal state for each clinical case. There are ten target

classes in this dataset. A deep neural network is trained on this dataset which gave an accuracy of 85% where the average recall and precision are 84.30 and 84.91%.

Zhao et al. [3] described a wearable device that uses accelerometers and machine learning to automatically detect fetal movement. The Internet of Things (IoT) is utilized on the system to connect all terminal monitoring devices to a control center, allowing the concept of e-health home care to be realized. The system is divided into two components: the local monitoring unit and the remote health evaluation unit. The dataset was collected by the authors on their own by collecting recordings from 14 pregnant women. This sound data was then pre-processed by filtering it through the IIR bandpass filter and then the feature extraction techniques used in this work are time-domain features and time-frequency features. The labeled dataset served as the training data for the machine learning classifier, which was subsequently used to process the newly given data. According to maternal markers, it has been noted that the system has correctly recognized fetal movements that are placed between 150 and 600 s. Wang et al. [4] have introduced fine-tuning an image-specific CNN model to make it adaptable to a particular image. BIFseg extracts the region inside the bounding box and feeds it once the user specifies a bounding box to the CNN model which are designed to learn some common features. The dataset in this work consists of images of brain and fetus which are used to train the CNN model and fine-tune it. This model includes the 2D segmentation of several organs from fetal MRI and the 3D segmentation of brain tumors. That included contrast-enhanced T1-weighted (T1c) and Fluid-attenuated Inversion Recovery (FLAIR) images used to verify the proposed framework. Finally, the refinement model, i.e., PC-net, was compared with other supervised models, in which PC-net gave the accepted results.

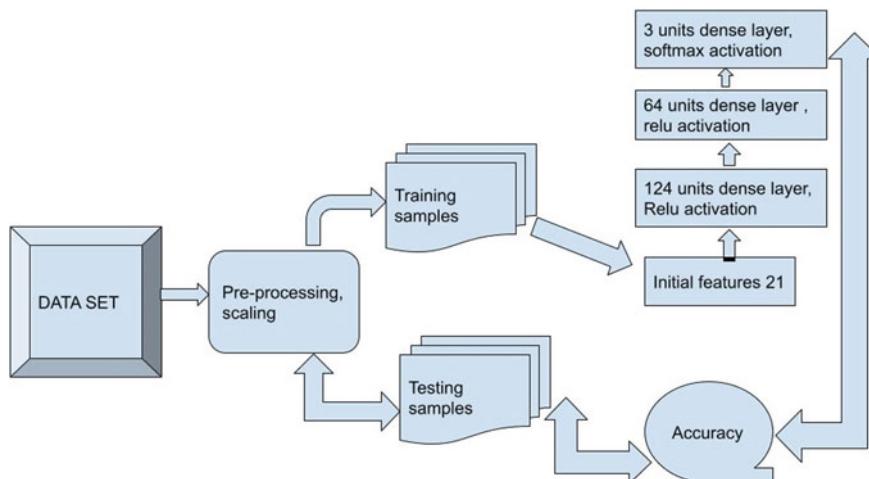
Jiaming et al. [5] have compared twelve machine learning models to classify the fetal health. The dataset used here is from the UCI machine learning repository it is the CTG dataset. To create a blender model, they combined the four top models using the soft-integration technique. The blender model performed excellently when compared to the simple machine learning models it gave an accuracy of 95%. Chin-naiyan et al. [6] have proposed a model that first trimester of pregnancy is the focus of the authors' investigation on fetal abnormalities. The major goal of this article review is to investigate the various machine learning processes for accurate diagnosis and prognosis of abdominal anomalies in order to lower the incidence rate. The numerous methods and studies used to accurately predict fetal health and development conditions from a set of pre-classified patterns are presented in this paper. The best model was ANN which gave an accuracy of 99.73%. Sahin et al. [18] have compared the categorization abilities of eight various machine learning techniques using ante partum cardiotocography (CTG) data. This study makes use of datasets from UCI that include CTG data with certain illustrative features. To select the most effective model from a variety of machine learning models, they were all compared. The model with the best accuracy was the random forest model with an accuracy of 99.18%.

Abbas et al. [19] compared the categorization abilities of eight various machine learning techniques using antepartum cardiotocography (CTG) data. This study makes use of datasets from UCI that include CTG data with certain illustrative

features. To select the most effective model from a variety of machine learning models, they were all compared. The random forest model provided the best accuracy which is 90%. Omneya et al. [20] aimed to classify the fetal brain abnormalities in early stages, before the fetal is born. The dataset used here consists of 227 fetal images between 1 and 39 weeks; this data was collected by the authors. The various feature extraction techniques used here are DWT and GLCM. The models used here to classify are LDA, Linear SVM, and KNN. Linear SVM gives the best accuracy of 74% before the enhancement using ROI, whereas Ensemble Subspace Discriminates gives the best accuracy of 80% after the enhancement. Syed Ahsin Ali et al. [21] aimed to classify the fetal distress using machine learning algorithms. The UCI machine learning repository's publicly accessible dataset was used in this study. In this study, the decision tree method was applied with a bagging strategy for complete features and pertinent features, this model gave an accuracy of 93.98%. Whereas when tested with random forest, it performed slightly better with an accuracy of 94.73%.

### 3 Methodology

This model is proposed based on the artificial neural network model with three layers. According to the model approach, initially, data we pre-processed the data with a min–max scale, then oversampling was done to balance all classes. Finally, the input was computed with a trained ANN model, and the performance of the model was evaluated using accuracy. Figure 1 shows the proposed model architecture.

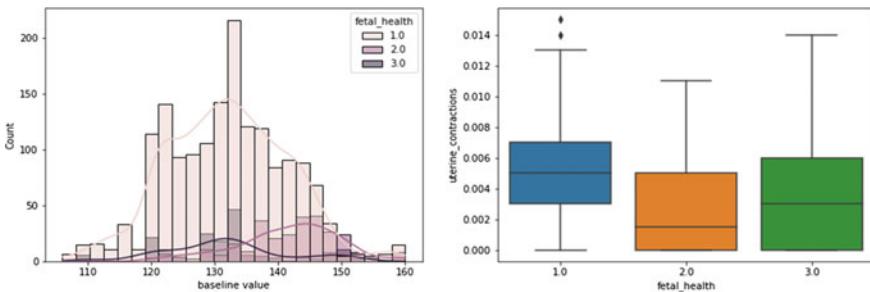


**Fig. 1** Architecture of proposed model

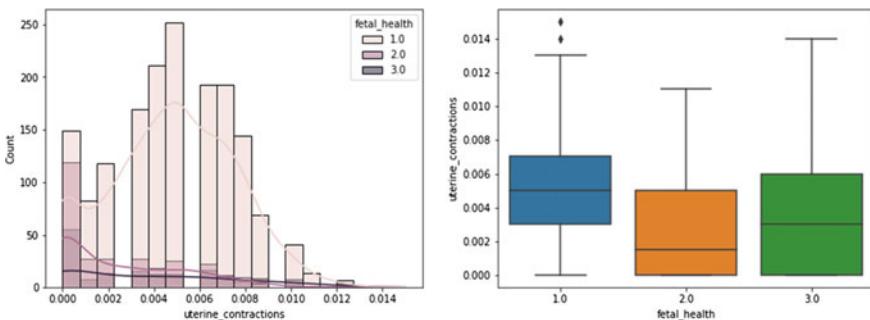
### 3.1 Dataset

Kaggle fetal health prediction [22] was used to conduct this research which has 21 features and one target variable, health status. After removing duplicate values, finally, it has 2126 cardiotocograms (CTG) samples of various types like fetal movement, accelerations, etc. The target values are normal, and Suspect and Pathological are labeled as 0, 1, 2. The distribution of data samples concerning fetal health is illustrated in Figs. 2 and Fig. 3 illustrate out layers of data over the classes.

From Fig. 4, it is observed that the dataset is imbalanced, i.e., the majority classes dominate the other features. Here to boost up the minority samples, SMOTE method used with a random sample of 42. The original number of samples is 2126; we generated 2839 synthetic samples to balance the data. The balanced data after SMOT is shown in Fig. 5. Furthermore, we did a min–max scalar to normalize the data. After the min–max scaling, we divide the data set into training and testing samples with a random size of 40 and a ratio of 75:25. Here, 75% will be the training data and 25% will be the testing data after SMOTE.

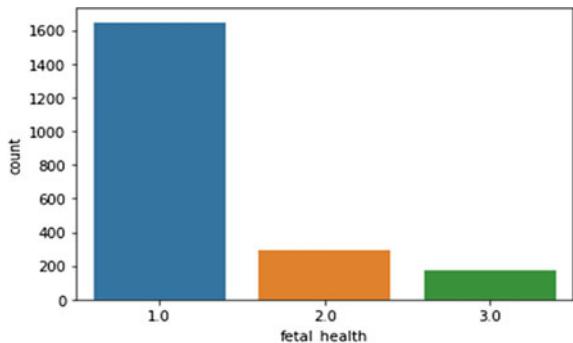


**Fig. 2** Bar chart and box plot for fetal health and base line values and uterine\_contractions

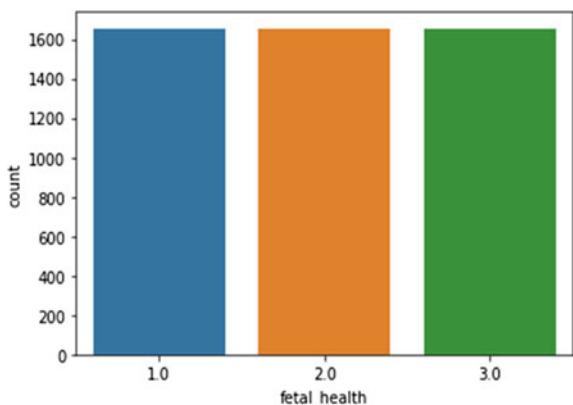


**Fig. 3** Bar chart and box plot for fetal health and base line values and uterine\_contractions

**Fig. 4** Imbalanced target values of fetal health data



**Fig. 5** Balanced data of fetal health data set after adding synthetic data



### 3.2 Implementation and Training

This research implemented a three-layer artificial neural network, with the input size of  $21*1$ , passed to an input layer  $124*1$  with ReLu activation function, and the output is passed to the next ANN layer of size  $64*1$ . Finally added a dense layer with size 3, activation function softmax. We used the Adam optimizer to update the weights and calculated categorical\_crossentropy loss. Table 1 presents the parameters used for proposed ANN model.

## 4 Result Analysis

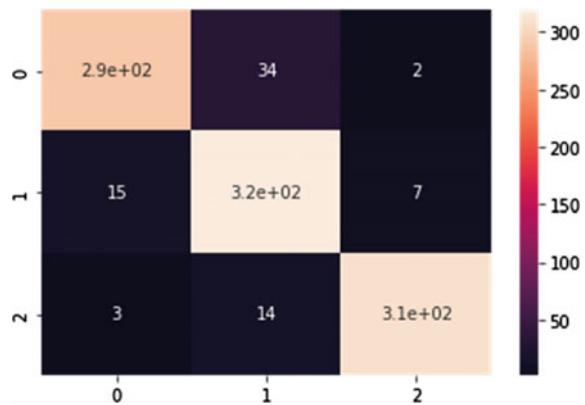
In this research, ANN model trained for 20 epochs and got an accuracy of 0.924. In our model, the training and validation accuracy both increased epoch by epoch consistently. Table 2 illustrates the final results of our proposed model. Figure 6 illustrates true positive and false negative results in a confusion matrix. Finally compared

**Table 1** Parameters used for ANN model

Parameters	Values
Learning rate	0.001
Input size	21*1
ANN layers	3 (two hidden layers)
Number of epochs	20
Activation	Relu, softmax
Optimizer	Adam
Loss	Categorical_crossentropy

**Table 2** Precision, recall, and F1-score for all three classes on balanced data set

	Precision	Recall	F1-score	support
Class 0	0.94	0.89	0.91	325
Class 1	0.87	0.94	0.90	341
Class 2	0.97	0.95	0.96	327
Accuracy			0.92	993
Macro avg	0.93	0.92	0.93	993
Weighted avg	0.93	0.92	0.92	993

**Fig. 6** Confusion matrix of proposed model

the results of actual (imbalanced) balanced data after adding synthetic data, and according to our observation the precision, recall, and F1-score of imbalanced data minority classes are 0.71, 0.75, and 0.73, respectively. After adding synthetic data at random samples, the minority classes' results like precision, recall, and F1-score of 0.87, 0.94, and 0.90 as given in Table 3. Before SMOTE, the minority classes are dominated by majority data.

**Table 3** Comparison of results on balanced and imbalanced data

	Without SMOTE			After SMOTE		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Class 0	0.91	0.92	0.92	0.94	0.89	0.91
Class 1	0.71	0.75	0.73	0.87	0.94	0.90
Class 2	0.81	0.68	0.75	0.97	0.95	0.96

## 5 Conclusion

From this research, we have introduced an ANN-based fetal health classification model that assists women in having healthy childbirth. In this approach, ANN has trained and attained classification accuracy of 92.0% over synthetic data obtained using SMOTE. This model is not overfitted and provides optimal accuracy. In our future work, we will include the dataset related to other health parameters, such as thyroid, blood sugar levels, blood pressure, that affect fetal health.

## References

1. Akbulut A, Ertugrul E, Topcu V (2018) Fetal health status prediction based on maternal clinical history using machine learning techniques. *Comput Methods Programs Biomed* 163:87–100
2. Miao JH, Miao KH (2018) Cardiotocographic diagnosis of fetal health based on multiclass morphologic pattern predictions using deep learning classification. *Int J Adv Comput Sci Appl* 9(5)
3. Zhao X, Zeng X, Koehl L, Tartare G, de Jonckheere J, Song K (2019) An IoT-based wearable system using accelerometers and machine learning for fetal movement monitoring. In: 2019 IEEE international conference on industrial cyber physical systems (ICPS). IEEE, pp 299–304
4. Wang G, Li W, Zuluaga MA, Pratt R, Patel PA, Aertsen M, Doel T et al (2018) Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE Trans Med Imaging* 37(7):1562–1573
5. Li J, Liu X (2021) Fetal health classification based on machine learning. In: 2021 IEEE 2nd international conference on big data, artificial intelligence and internet of things engineering (ICBAIE). IEEE, pp 899–902
6. Chinnaiyan R, Alex S (2021) Machine learning approaches for early diagnosis and prediction of fetal abnormalities. In: 2021 international conference on computer communication and informatics (ICCCI). IEEE, pp 1–3
7. Mehbodniya A, Prabhu Lazar AJ, Webber J, Sharma DK, Jayagopalan S, Singh P, Rajan R, Pandya S, Sengan S (2022) Fetal health classification from cardiotocographic data using machine learning. *Expert Syst* 39(6):e12899
8. Fung R, Villar J, Dashti A, Ismail LC, Staines-Urias E, Ohuma EO, Salomon LJ et al (2020) Achieving accurate estimates of fetal gestational age and personalised predictions of fetal growth based on data from an international prospective cohort study: a population-based machine learning study. *The Lancet Digital Health* 2(7):e368–e375
9. Tao J, Yuan Z, Sun L, Yu K, Zhang Z (2021) Fetal birthweight prediction with measured data by a temporal machine learning method. *BMC Med Inf Dec Mak* 21(1):1–10
10. Pu B, Li K, Li S, Zhu N (2021) Automatic fetal ultrasound standard plane recognition based on deep learning and IIoT. *IEEE Trans Industr Inf* 17(11):7771–7780

11. Xie HN, Wang N, He M, Zhang LH, Cai HM, Xian JB, Lin MF, Zheng J, Yang YZ (2020) Using deep-learning algorithms to classify fetal brain ultrasound images as normal or abnormal. *Ultrasound Obstet Gynecol* 56(4):579–587
12. Kannan E, Ravikumar S, Anitha A, Kumar SAP, Vijayasarathy M (2021) Analyzing uncertainty in cardiotocogram data for the prediction of fetal risks based on machine learning techniques using rough set. *J Ambient Intell Humanized Comput* 1–13
13. Guedalia J, Lipschuetz M, Novoselsky-Persky M, Cohen SM, Rottenstreich A, Levin G, Yagel S, Unger R, Sompolinsky Y (2020) Real-time data analysis using a machine learning model significantly improves prediction of successful vaginal deliveries. *Am J Obstet Gynecol* 223(3):437-e1
14. Scheinost D, Pollatou A, Dufford AJ, Jiang R, Farruggia MC, Rosenblatt M, Peterson H et al (2022) Machine learning and prediction in fetal, infant, and toddler neuroimaging: a review and primer. *Biol Psychiatry*
15. Alam MT, Islam Khan MA, Dola NN, Tazin T, Khan MM, Albraikan AA, Almalki FA (2022) Comparative analysis of different efficient machine learning methods for fetal health classification. *Appl Bion Biomech*
16. Agrawal K, Mohan H (2019) Cardiotocography analysis for fetal state classification using machine learning algorithms. In: 2019 International conference on computer communication and informatics (ICCCI). IEEE, pp 1–6
17. Hussain M, Nadia, AUR, Ben Othman MT, Zafar J, Zafar H, Hamam H (2022) Accessing artificial intelligence for fetus health status using hybrid deep learning algorithm (AlexNet-SVM) on cardiotocographic data. *Sensors* 22(14):5103
18. Sahin H, Subasi A (2015) Classification of the cardiotocogram data for anticipation of fetal risks using machine learning techniques. *Appl Soft Comput* 33:231–238
19. Abbas R, Hussain AJ, Al-Jumeily D Baker T, Khattak A (2018) Classification of fetal distress and hypoxia using machine learning approaches. In: International conference on intelligent computing. Springer, Cham, pp 767–776
20. Attallah O, Gadelkarim H, Sharkas MA (2018) Detecting and classifying fetal brain abnormalities using machine learning techniques. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA). IEEE, pp 1371–1376
21. Shah SAA, Aziz W, Arif M, Nadeem MSA (2015) Decision trees based classification of cardiotocograms using bagging approach. In: 2015 13th international conference on frontiers of information technology (FIT). IEEE, pp 12–17
22. <https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification>. Accessed on 5 Jan 2023

# Cryptonate: Crypto-Jacking Web Extension



Nilesh Patil, Dhruv Gandhi, Pranay Prajapati, and Kevin Haria

**Abstract** Cybercriminals are quickly shifting their focus from ransomware to crypto-jacking due to the emergence of numerous cryptocurrencies and their rising value, as well as the lower risk and greater potential for financial benefit. Using infected electronic devices and networks, crypto-jacking enables attackers to mine Bitcoins in a less complex and less observable manner than ransomware attacks. The paper presents a novel approach to detect and prevent crypto-jacking by utilizing a combination of static and dynamic methods. The static approach utilizes a black-list of known malicious websites and disables JavaScript on these sites to prevent unauthorized mining. The dynamic approach employs a machine learning model to identify any signs of malicious activity. If such activity is detected, the website is added to the blacklist and JavaScript is disabled to protect the device's resources. This solution not only provides a proactive approach to detecting and preventing crypto-jacking but also offers a multi-layered defense mechanism by combining both static and dynamic methods. The use of machine learning for real-time monitoring and detection of crypto-jacking activity offer a more sophisticated solution than traditional blacklisting methods. The proposed solution has the potential to be a valuable tool in the fight against the growing threat of crypto-jacking.

**Keywords** Cryptocurrency · Blockchain · Crypto-mining · Crypto-jacking

## 1 Introduction

In this section, we will discuss the fundamentals of cryptocurrency, crypto-mining, and crypto-jacking.

---

N. Patil (✉) · D. Gandhi · P. Prajapati · K. Haria

Department of Computer Engineering, SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai 400056, India

e-mail: [nileshdeep@gmail.com](mailto:nileshdeep@gmail.com)

## ***1.1 Cryptocurrency***

Cryptocurrency is a digital or virtual currency that uses cryptography for security and operates independently of a central bank. Cryptocurrencies are decentralized systems that allow for the transfer of funds directly between individuals, without the need for intermediaries such as banks. Cryptocurrency values have made them popular as trading and investing instruments. To a limited extent, they are also used for cross-border transfers [1]. The most well-known cryptocurrency is Bitcoin, but there are thousands of other cryptocurrencies in existence, each with its own unique features and applications. Some cryptocurrencies are used primarily as investments, while others are designed to provide a more secure and efficient means of payment [2]. Cryptocurrencies use complex mathematical algorithms and blockchain technology to securely record transactions on a decentralized ledger, ensuring that the same unit of currency cannot be spent twice by the same user. This makes cryptocurrency transactions extremely secure and tamper-proof [3].

## ***1.2 Crypto-Mining***

Crypto-mining, also known as cryptocurrency mining, is the process of verifying transactions on a blockchain network and adding them to the public ledger. In return for their effort, miners are rewarded with a certain number of newly created cryptocurrency units. Crypto-mining requires powerful computers to solve complex mathematical problems, which are used to validate transactions and create new blocks in the blockchain. As more blocks are added to the chain, the difficulty of the problems increases, making it more difficult and resource-intensive to mine for cryptocurrency [4]. The process of crypto-mining is an essential component of many cryptocurrency systems, as it helps to ensure the security and integrity of the blockchain network by making it more difficult for malicious actors to manipulate the ledger [5]. While some people mine cryptocurrency as a hobby or for personal gain, others engage in it as a profitable business. However, crypto-mining can be a resource-intensive process that requires a lot of electricity and specialized hardware, so it may not be cost-effective for everyone.

## ***1.3 Crypto-Jacking***

Crypto-jacking is the unauthorized use of someone else's computer resources to mine cryptocurrency. This is usually done by installing malicious software on a target's computer that runs in the background, using the computer's processing power to mine for cryptocurrency without the user's knowledge or consent. Crypto-jacking can have a significant impact on a computer's performance, as it can slow down the

system and increase the wear and tear on the hardware. In some cases, it can also cause the computer to crash or become unresponsive. Crypto-jacking is becoming an increasingly common problem, as the value of some cryptocurrencies has risen dramatically in recent years. Attackers are attracted to this form of hacking because it can be a lucrative and low-risk way to mine for cryptocurrency, as the costs of the computing resources are borne by the victims rather than the attacker [6]. To protect against crypto-jacking, it is important to keep your computer and software up-to-date with the latest security updates and to use anti-malware and anti-virus software to detect and remove any malicious software that may be installed on your computer [7].

This paper is structured as follows: Sect. 2 provides details on the literature survey. Section 3 explains the related work already done in detection and prevention of crypto-jacking. In Sect. 4 the details about the methods we have used in our project are present. Section 5 showcases our work. Section 6 gives the result of the work done. Section 7 provides the conclusion.

## 2 Literature Survey

Crypto-jacking is a type of cyberattack that involves the unauthorized use of someone else's computing resources, such as CPU and memory, to mine for cryptocurrencies. The process of mining cryptocurrencies involves solving complex mathematical problems in order to validate transactions and earn rewards in the form of new coins. Crypto-jacking attacks typically occur through the use of malicious software that is downloaded onto a device without the user's knowledge or consent. This software can come in the form of malware, browser extensions, or even through a website's JavaScript code. Once the software is installed, it will use the victim's device to mine for cryptocurrencies, consuming significant amounts of computing resources and slowing down the device. The miner will then keep the rewards earned from the mining process. Crypto-jacking has become a popular form of attack due to the rising value of cryptocurrencies and the ease of launching a successful attack. Unlike other types of cyberattacks, such as ransomware, crypto-jacking is often difficult to detect and can persist for an extended period of time, silently stealing valuable computing resources and earning the attacker a profit. It is important for individuals and organizations to take steps to protect against crypto-jacking, such as keeping the software and operating systems up-to-date, using anti-virus software and firewalls, and being cautious of suspicious downloads and emails.

Bitcoin, which was introduced in 2009 as an open-source software by Satoshi Nakamoto, is recognized as the first decentralized cryptocurrency [8]. Since then, many cryptocurrencies were introduced into the market (Bytecoin, Monero, Dogecoin, Litecoin, etc.), and the prevalence of such cryptocurrency kept on growing. The growth of prevalence and prices of cryptocurrencies in the market has attracted many cybercriminals. Cybercriminals are trying all the possible ways to mine the

cryptocurrencies and gain monetary benefits. One such type of attack used by cyber-criminals to gain benefits is through browser-based crypto-jacking. Our focus in the current research paper is on most prevalent cryptocurrency that is used in the browser-based crypto-jacking, namely, Bitcoin, Bytecoin, and Monero. Crypto-jacking using browser as a medium is a novel attack strategy and an emerging business model that has not been investigated completely.

## 2.1 *Crimes Committed Using Crypto-Jacking*

The cryptocurrency-related attacks have been prevalent in the recent time, and companies that cater to a large user group have been targeted. The SDK of a web chat system LiveHelpNow was targeted by the cybercriminals which resulted into illicit mining across 1500 websites [9]. Another such attack was on the PolitiFact, a political fact checking website which was injecting crypto-jacking code [10, 11]. The governmental organizations of the UK such as Information Commissioners Office, Manchester city Council, and 4200 more websites were attacked by the injection of crypto-jacking script through the website plugin Browsealoud [12]. Tesla, Showtime, and LA Times website are other such companies whose website was affected by the crypto-jacking [13, 14]. The WannaCry ransomware attack of 2017 used cryptocurrency to infect up to 300,000 victims from 150 countries where the aim was to extort money in the form of cryptocurrencies [15].

## 2.2 *Background Analysis of Crypto-Jacking Prevalence*

In [16] and [17], it has been found that in Alexa ranking, one out of 500 websites contains web-based miner that immediately starts mining activities when visited. The target cryptocurrencies are Monero, Bytecoin, and Electroneum. The implementation is done using CoinHive miner and with minor modification can support different currencies. The [17] identifies 4627 suspicious sites in the Alexa ranking of which 42% of websites are active crypto-jacking sites. The paper also highlights that crypto-jacking is indeed a widespread phenomenon not limited to popular websites while examining top 1 million sites in the Alexa ranking. The paper shows that top crypto-jacking websites are hosted by the USA, followed by Russia, Germany, France, the Netherlands, and others. The content-based distribution of crypto-jacking websites by the paper indicates that Entertainment, Pornography, Business, Education, News, Games, and TV/Video streams are in the list of top ten categories that are prone to carry out crypto-jacking activity. A similar independent content-based classification done by [18] indicates similar results where the major prevalence was found in website containing content of category entertainment, business, education, and adult. In [17], an approach to investigate the crypto-jacking phenomenon using a three-phase analysis approach has been proposed. The [18] shows that top five top-level

domains (TLDs) of crypto-jacking websites are .com, .net, .si, .online, and .ru. Rüth et al. [20] found that 137 million domains out of the 138 million domains are of com/net/org domains, out of which 1 million belong to the Alexa's top 1 million list. The author finds 0.08% of the analyzed set carry out browser-based crypto-jacking. Hong et al. [19] made an in-depth analysis of 853,936 popular webpages where 868 unique crypto-jacking samples from a set of 2770 unique samples belong to the Alexa's top 100 K ranking websites. Konoth et al. [21] conducted an analysis of websites affected by drive-by mining within the top 1 million websites ranked by Alexa. They also examined the various techniques employed to avoid detection.

### 2.3 *Detection of Crypto-Jacking*

Many detection methodologies have been proposed for detection of in-browser crypto-jacking. Our approach is very similar to the one mentioned in [22]. They suggested a network traffic analysis-based detection mechanism which works even on encrypted network traffic. The normal and VPN encrypted network traffic of three cryptocurrencies is taken into account, namely Bitcoin, Bytecoin, and Monero. They also propose a machine learning (ML)-based system named Crypto-Aegis that can detect cryptocurrency -elated behaviors such as pool and solo mining, and active full node on the basis of findings from the analysis. The model achieves a F1-score of 0.96 and AUC-ROC of 0.99 along with benefits of being device and infrastructure in-dependency. However, the approach proposed requires constant monitoring of network traffic which is resource-intensive task in a commercial scenario and may require establishment of extra hardware and monitoring tool or third party paid monitoring software to handle the corporate scenario network traffic. Another machine learning-based detection mechanism is proposed in [23] where a CapJack, named detector, uses the CapsNet machine learning algorithm and system features such as CPU, memory, disk, and network utilization to detect crypto-mining activities. The host-based solution proposed has a success rate of 87%. The proposed approach requires constant monitoring of resources and thus incurs unnecessary load on the host computer because the instances of crypto-jacking attack are rare. Ning et al. [24] proposed a hybrid approach to detect the crypto-jacking using a browser extension named CMBLOCK having the static blacklisting and dynamic behavior-based analysis feature for the detection of crypto-jacked websites. The behavior-based analysis that has been proposed is susceptible to the hidden library call and code obfuscation. Razali et al. [25] proposed a dynamic machine learning technique based on op-code analysis for browser-based crypto-mining detection. The op-code analysis was successful in classifying the crypto-mining sites, benign websites with injected crypto-mining code, in-active crypto-mining website, and actual benign website. This approach too is uncalled for as it is resource-intensive task as well as the crypto-jacking events are rare compared to day-to-day browsing activities. Some static existing browser-level countermeasures are web extensions such as No Coin, Anti Miner, and No Mining in [26, 27], and [28] which utilize blacklisting approach

for detection, but it can be easily bypassed by adaptive attacker through creating new links (proxy). In [29], a framework called BMDetector has been proposed for detection of in-browser crypto-mining by analyzing browser heap snapshot and stack data once the hooking of JavaScript in the kernel source of Chrome Webkit is done. The model used is a Recurrent Neural Network (RNN), when applied to 1159 samples, the detection rate was at 98% for samples which are not encrypted and 92% otherwise. In [30], a dynamic approach, Secure In-lined Script Monitors for Interrupting CryptoJacks (SEISMIC), which focuses on hard to obfuscate semantic features for detection, has been introduced. The analysis Wasm script is done, and based on it, a statistical model of mining and non-mining behavior is extracted. The implementation of the detection mechanism was done using support vector machine which gave an accuracy of 98% or above when evaluated using tenfold cross-validation.

## 2.4 Monetary Benefits of Crypto-Jacking

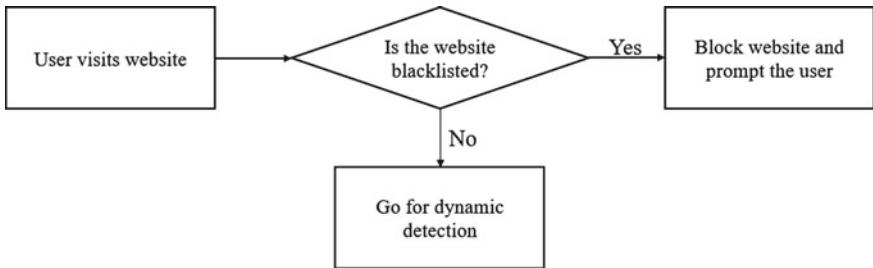
In-browser crypto-mining is debated as an alternative to advertisement. M. Saad et al. [18] showed that the revenue earned through the crypto-jacking is negligible compared to the revenue earned through online advertisements. M. Musch et al. [17] used a statistical approach to determine the revenue generated by active miners. The author finds that most profitable sites are able to generate 119–340 USD which is quite a moderate revenue. A similar statistical estimation is done in [31], where they analyzed a domain parking service for a period of 3 months and found out that for the given period the revenue generated would be 7.69 USD.

## 3 Methodology

Our product is a web extension, which works in the background while the user is surfing on the web. The web extension checks for possible crypto-jacking on users' computing devices. The following two different methods are used:

### 3.1 Static Approach

The initial check is done by the blacklisting method, and if a particular website is on the blacklist, then the next dynamic check is performed to confirm the crypto-jacking activities. If the website is performing illicit crypto-jacking activities, then the web extension will enlighten the user about the crypto-jacking activity, and a predefined action is undertaken, such as blocking the website or disabling the scripts as depicted in Fig. 1. If the website is performing non-illicit crypto-jacking activities, then the web extension will enlighten the user about the crypto-jacking activity and

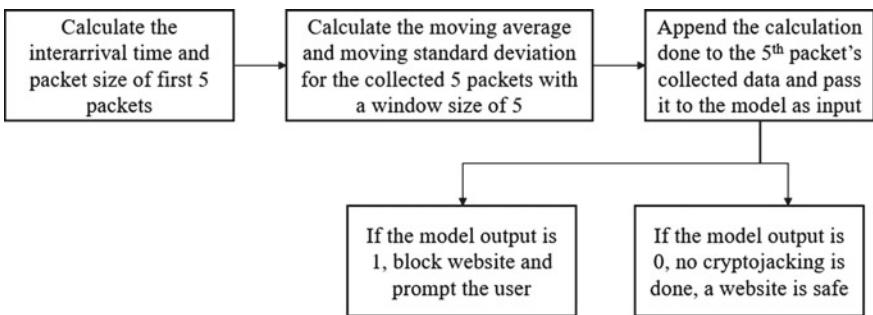


**Fig. 1** Static approach of crypto-jacking

what possible damage to the resources can occur due to it. If the user permits the use of its resources for crypto-jacking activities, then no actions are to be taken by the web extension, else the predefined action against that particular website is taken. The initial level of defense against the crypto-jacking is the static-based black listing. This level detection is only used to check for possibility of crypto-jacking but does not guarantee whether the crypto-jacking is actually performed by the website or not. This means that websites that previously deployed illicit crypto-mining on their website have either removed the crypto-mining code or the crypto-mining code currently present is not active.

### 3.2 Dynamic Approach

The second level of defense is the method of monitoring the network traffic as shown in Fig. 2. Based on the nature of network traffic and pattern, decision is made whether or not the website illicitly mining crypto currencies. Our proposed design does not undergo rigorous dynamic approach but only undertake one based on whether that particular website is black listed or not or whether that particular website is connected to some other black listed website.



**Fig. 2** Dynamic approach of crypto-jacking

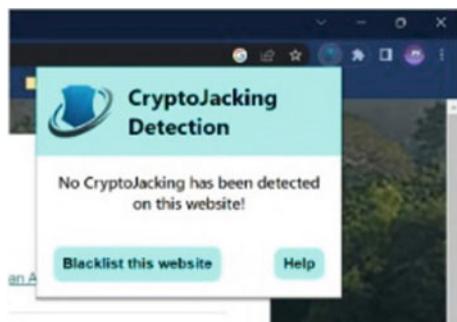
## 4 Proposed Work

In this section, we discuss the methodology used for crypto-jacking using both the static approach and the dynamic approach.

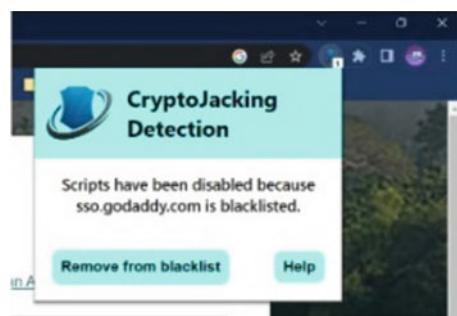
### 4.1 Static Approach

For the static method, we have used a text file which contains a large number of links of the websites that are identified as potentially crypto-jacking websites. Whenever user visits a website, it is checked with the already identified websites; if it is already identified as crypto-jacking website, then the user is notified about it. The results are shown in Figs. 3 and 4, respectively.

**Fig. 3** Manually blacklisting website



**Fig. 4** Blacklisted website



**Table 1** Benign website network traffic dataset statistic

Statistic/feature	Interarrival time (s)	Packet size (bytes)
Mean	6.835	373.033
Median	0.894	182
Count	921,067	

**Table 2** Crypto-jacking website network traffic dataset statistic

Statistic/feature	Interarrival time (s)	Packet size (bytes)
Mean	0.0224	770.138
Median	0.00403	835
Count	12,651	

## 4.2 Dynamic Approach

The dynamic approach is followed when the visited website is not present in the blacklist. The network traffic is monitored dynamically for any potential crypto-mining activity. This monitoring starts as soon as it is determined that the website is not present in the blacklist. The monitoring for that respective website is done until at least 15–20 ingoing packets are obtained. These packets are then normalized and standardized before they are sent to the machine learning model for the prediction. If any of the prediction is positive for crypto-jacking, then the user is informed of illicit crypto-mining activity. The description of the model preparation is explained in the following sub-sections.

### 4.2.1 Dataset

The dataset used consists of interarrival time and packet size. The crypto-mining network traffic are of most prevalent cryptocurrency that are deployed for in-browser crypto-jacking which are Bytecoin, Bitcoin, and Monero. The network traffic of benign website is a normal traffic of random browsing. The dataset contains in total of 933,718 packets. The dataset is quite imbalance with 921,067 packets collected from benign websites network traffic and a total of 12,651 packets from crypto-mining website network traffic. The statistic of the collected network traffic is displayed in Tables 1 and 2, respectively. The statistic shows that crypto-mining network traffic consists of larger ingoing packets, and it was found that the interarrival time of those packets is comparatively less.

### 4.2.2 Sampling Method Selection, Training, and Testing

To handle the imbalance nature of the dataset, several sampling methodologies were taken into consideration to select the best sampling strategy to be used for building the

final model. But before carrying out any of the sampling methodologies, the dataset was initially split into 60% for training and 40% for testing. The various sampling strategies are then applied to the training set only. The sampling strategies taken into consideration are under-sampling methods such as Random, Tomek Links, Neighborhood cleaning rule, and Near miss; oversampling methods such as Random, Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic (ADASYN), and Borderline Synthetic Minority Oversampling Technique (SMOTE); and an over-under sampling method such as SMOTE Tomek. The classifier used for the detection is the XGBoost Classifier. The following are the reasons for selecting the XGBoost Classifier in comparison to other complex machine learning algorithms:

1. It has an internal regularization parameter to avoid over-fitting.
2. It is designed to be highly scalable and can handle large datasets with millions of rows and thousands of columns.
3. It can handle time series data (such as network traffic) better than other complex algorithms (other bagging and boosting machine learning algorithms).

The classifier consists of 136 boosting rounds where in each round 25 trees are built in parallel with each of maximum depth of 6. Once the base classifier's parameters are set, then for each sampling method the classifier is trained on the obtained results of applying the sampling method on the training set, and later the trained classifier is tested against the testing set. The results of each sampling are evaluated and the best sampling method and the classifier trained on these samples are used as the final model in our web extension.

From Table 3, we can see that the Tomek Links is the best sampling method to be used for creating the sample for training the classifier as it gives a better F1-score (0.90), accuracy (99.363%), and balanced accuracy (97.625%). The model built on the sample generated from Tomek Links had an AUC of the ROC of 0.98 as depicted in Fig. 6. The confusion matrix generated shows that the model performs efficiently on the unseen dataset making it able to classify the packets generated crypto-mining and benign network traffic. The confusion matrix is shown in Fig. 5.

## 5 Novelty

In this section, we compare our approach with the existing approaches and also discuss the novel features of the proposed system.

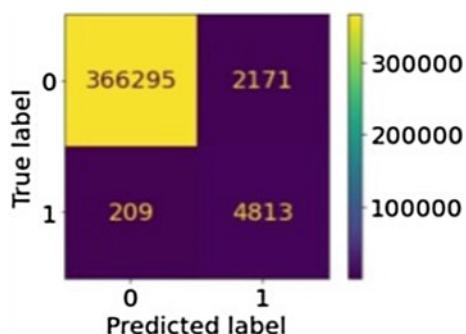
### 5.1 Comparative Analysis with the Existing Standard Approach

The approach proposed is a two-layered detection mechanism where the layer 1 is of static blacklisting method and layer 2 is of dynamic network traffic-based analysis.

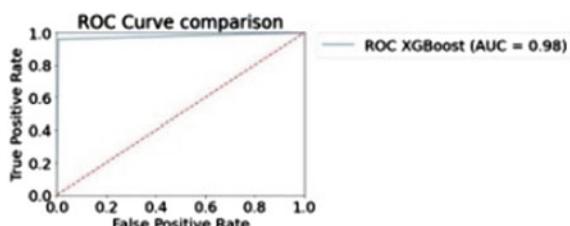
**Table 3** Test results obtained corresponding to the model trained on the sample obtained after applying the respective sampling method

Sampling method	Test results				
	Precision (macro-average)	Recall (macro-average)	F1-score (macro-average)	Accuracy (%)	Balanced accuracy (%)
Random undersampling	0.74	0.99	0.82	98.508	98.792
Tomek Links	0.84	0.98	0.90	99.363	97.625
Neighborhood cleaning rule	0.83	0.98	0.90	99.160	96.983
Near miss	0.52	0.81	0.41	62.053	80.581
Random oversampling	0.77	0.99	0.85	98.859	98.950
SMOTE	0.77	0.99	0.85	98.846	98.738
ADASYN	0.74	0.97	0.81	98.512	96.624
Borderline SMOTE	0.71	0.97	0.79	98.191	97.374
SMOTE Tomek	0.77	0.99	0.85	98.865	98.747

**Fig. 5** Confusion matrix of XGBoost Classifier built with Tomek Link under sampling



**Fig. 6** ROC curve and related AUC of XGBoost Classifier built with Tomek Link under sampling



**Table 4** Comparison of crypto-jacking approaches

Approach proposed in [22]	Our proposed approach
Single-layer detection mechanism	Two-layer detection mechanism
Requires constant monitoring of network traffic for crypto-jacking detection	Requires network traffic monitoring of only those websites not present in blacklist
Model used for detection is deployed on different server	Model used for detection is deployed on the same host machine
Requires additional hardware and software for deployment	Uses the available host resources
Machine learning algorithm used is Random Forest	Machine learning algorithm used is XGBoost Classifier

Most of the standard approaches use either static or dynamic approach for detection. The detection mechanism such as [24] uses the two-layered in which the dynamic approach is susceptible to code obfuscation and hidden library call. The network traffic analysis done in layer 2 is of the proposed detection mechanism which is immune to hidden library call and code obfuscation. The approach proposed reduces the need of constant network-based traffic monitoring for detection by utilizing the blacklist as the base to avoid the network-based monitoring of the website that are already in the blacklist.

## 5.2 Comparative Analysis with Approach Proposed in [22]

The proposed approach in [22] requires constant monitoring of the network traffic for detection mechanism where model-trained Random Forest machine learning algorithm is used. Our approach, in contrast, is based on a two-layered detection mechanism. The network-based detection is employed only when the website under analysis is not present in the blacklist. The approach proposed here can be deployed directly on host machine, whereas the approach in [22] requires separate server for deployment. The model used for network traffic classification is trained using XGBoost Classifier. The approach proposed here requires no additional hardware or software for detection, whereas it utilizes the available host hardware and software. The detailed point of differences is mentioned in Table 4.

## 6 Results and Discussion

It was found that the mean interarrival time of crypto-jacking website network trace was 0.0224 s and that of mean interarrival time of benign website network trace was 6.835 s. This shows that there is a significant difference between mean interarrival time of crypto-jacking website and the benign website. The mean packet size of

crypto-jacking website network trace was 770.138 bytes and that of mean packet size of benign website network trace was 373.033 bytes. This shows that there is significant difference between mean packet size of crypto-jacking website and the benign website. Our project achieves a F1-score of 0.90, recall of 0.98, and 0.98 of AUC for the ROC curve. Our project reduced the frequency of need of dynamic detection and prevention of crypto-jacking by 65%, thus making detection and prevention of crypto-jacking faster. Since we first check the website with a blacklist already present, there is no need of using dynamic method unless no result is obtained using blacklist thus saving time.

## 7 Conclusion

In this paper, we took a systematic look at in-browser crypto-jacking through static analysis and dynamic analysis. In order to that, we have collected a dataset of crypto-jacking websites and performed static analysis that can be used to detect crypto-jacking from malicious and benign websites. We explore, through dynamic analysis, how in-browser crypto-jacking uses packets from the network and use that knowledge to reconstruct the operation of crypto-jacking. By surveying prior countermeasures and examining their limitations, we highlight long-term solutions, capitalizing on the insights from our static and dynamic analyses. The proposed two-layered architecture portrays that a hybrid detection mechanism can be used to overcome the limitations of individual approaches for the detection of in-browser crypto-jacking combining the accuracy of dynamic approach and resource-friendly static approach. Future work related to this must focus on the combination of other static and dynamic approaches for detection and a multi-layered architecture should also be explored.

## References

1. <https://www.investopedia.com/terms/c/cryptocurrency.asp>. Accessed 11 Feb 2022
2. <https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency/>. Accessed 8 Feb 2022
3. <https://www.coinbase.com/learn/crypto-basics/what-is-mining/>. Accessed 28 Jan 2022
4. <https://www.investopedia.com/tech/how-does-bitcoin-miningwork/>. Accessed 8 Feb 2022
5. <https://corporatefinanceinstitute.com/resources/cryptocurrency/bitcoinmining/>. Accessed 16 Feb 2022
6. <https://www.outlookindia.com/business/crypto-jacking-cases-arising-globally-why-so-and-should-this-worry-you-news-212990/>. Accessed 9 Feb 2022
7. <https://www.imperva.com/learn/application-security/crypto-jacking/>. Accessed 28 Jan 2022
8. Nakamoto S (2022) Bitcoin: a peer-to-peer electronic cash system, cryptography mailing list at metzdowd.com, <https://bitcoin.org/bitcoin.pdf>. Accessed 22 Apr 2022
9. LiveHelpNow. Security incident Nov 23rd, 2017. <https://livehelpnow.net/blog/security-incident-nov-23rd-2017/>. Accessed 23 Apr 2022

10. WallStreetJournal. Your computer may be making bitcoin for hackers <https://www.wsj.com/articles/hackers-latest-move-using-your-computer-to-mine-bitcoin-1509102002>. 2017. Accessed 23 Apr 2022
11. Washingtonpost. Hackers have turned politifact's website into a trap for your pc. <https://www.washingtonpost.com/news/theswitch/wp/2017/10/13/hackers-have-turned-politifacts-websiteinto-a-trap-for-your-pc/>. Accessed 23 Apr 2022
12. The Register. UK ICO, USCourts.gov... Thousands of websites hijacked by hidden crypto-mining code after popular plugin pwned. [https://www.theregister.com/2018/02/11/browsealoud\\_compromised\\_coinhive/](https://www.theregister.com/2018/02/11/browsealoud_compromised_coinhive/). Accessed 1 May 2022
13. NakedSecurity. Unsecured aws led to crypto-jacking attack on la times. <https://nakedsecurity.sophos.com/2018/02/27/unsecuredaws-led-to-crypto-jacking-attack-on-la-times/>. Accessed 1 May 2022.
14. TheVerge. Showtime websites secretly mined user cpu for cryptocurrency. <https://www.theverge.com/2017/9/26/16367620/showtime-cpu-cryptocurrency-monero-coinhive>. Accessed 1 May 2022
15. Paquet-Clouston M, Haslhofer B, Dupont B (2019) Ransomware payments in the bitcoin ecosystem. J Cybersecur 5(1):tyz003. <https://academic.oup.com/cybersecurity/article/5/1/tyz003/5488907>. Accessed 12 May 2022
16. van Saberhagen N (2022) Cryptonote v2.0. Technical report, CryptoNote, Oct. 2013. <https://bytecoin.org/old/whitepaper.pdf>. Accessed 15 May 2022
17. Musch M, Wressnegger C, Johns M, Rieck K (2019) Thieves in the browser: Web-based crypto-jacking in the wild. In: Proceedings of the 14th international conference on availability, reliability and security, in: ARES '19, ACM, New York, NY, USA, pp 4:1–4:10. <https://doi.org/10.1145/3339252.3339261>
18. Saad M, Khormali A, Mohaisen A (2018) End-to-End Analysis of In-Browser Crypto-jacking, Cornell University, Computer Science, Cryptography and Security. <https://arxiv.org/abs/1809.02152>
19. Hong G, Yang Z, Yang S, Zhang L, Nan Y, Zhang Z, Yang M, Zhang Y, Qian Z, Duan H (2018) How you get shot in the back: a systematical study about crypto-jacking in the real world. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, ACM, pp 1701–1713. <https://doi.org/10.1145/3243734.3243840>
20. Rüth J, Zimmermann T, Wolsing K, Hohlfeld O (2018) Digging into browser-based crypto mining. In: Proceedings of the internet measurement conference 2018, in: IMC '18, ACM, New York, NY, USA, pp 70–76. <https://doi.org/10.1145/3278532.3278539>
21. Konoth RK, Vineti E, Moonsamy V, Lindorfer M, Kruegel C, Bos H, Vigna G (2018) Minesweeper: an in-depth look into drive-by cryptocurrency mining and its defense. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, CCS '18, ACM, New York, NY, USA, pp 1714–1730. <https://doi.org/10.1145/3243734.3243858>
22. Caprolu M, Raponi S, Olieri G, di Pietro (2019) Cryptomining makes noise: a machine learning approach for crypto-jacking detection. <https://doi.org/10.1016/j.comcom.2021.02.016>
23. <https://www.sciencedirect.com/science/article/pii/S0140366421000797?via%3Dhub>
24. Ning R, Wang C, Xin C, Li J, Zhu L, Wu H (2019) Capjack: capture in-browser crypto-jacking by deep capsule network through behavioral analysis. In: IEEE INFOCOM 2019—IEEE conference on computer communications, pp 1873–1881. [https://doi.org/10.1109/ICINFOCOM4815-15.52854\\_13](https://doi.org/10.1109/ICINFOCOM4815-15.52854_13)
25. Razali MA, Mohd Shariff S (2019) Cmblock: in-browser detection and prevention crypto-jacking tool using blacklist and behavior-based detection method. In: Badioze Zaman H, Smeaton AF, Shih TK, Velastin S, Terutoshi T, Mohamad Ali N, Ahmad MN (eds) Advances in visual informatics, Springer International Publishing, Cham, pp 404–414. [https://doi.org/10.1007/978-3-030-34032-2\\_36](https://doi.org/10.1007/978-3-030-34032-2_36)
26. Carlin P, O’Kane S, Sezer J (2018) Burgess, detecting cryptomining using dynamic analysis. In: 2018 16th annual conference on privacy, security and trust (PST), pp 1–6. <https://ieeexplore.ieee.org/document/8514167>

27. Keramidas R (2022) Nocoin, Feb 2018. Available: <https://github.com/keraf/NoCoin>. Accessed 2 June 2022
28. Tunghobrens (2018) Anti miner–coin minerblock. <https://chrome.google.com/webstore/detail/miner-blocker-blockcoin/ejpcojkcallnhphimknkaoojohidegf>
29. Mining N (2018) Secure your browser. [https://www.hugedomains.com/domain\\_profile.cfm?d=nomining.com](https://www.hugedomains.com/domain_profile.cfm?d=nomining.com). Accessed 18 July 2022
30. Liu J, Zhao Z, Cui X, Wang Z, Liu Q (2018) A novel approach for detecting browser-based silent miner. In: 2018 IEEE third international conference on data science in cyberspace (DSC), pp 490–497. <https://ieeexplore.ieee.org/document/8411900/>. Accessed 18 July 2022
31. Wang W, Ferrell B, Xu X, Hamlen KW, Hao S (2018) Seismic: secure in-lined script monitors for interrupting cryptojackers. In: European symposium on research in computer security ESORICS 2018: computer security, pp 122–142. [https://doi.org/10.1007/978-3319-98989-1\\_7](https://doi.org/10.1007/978-3319-98989-1_7)
32. Eskandari S, Leoutsarakos A, Mursch T, Clark J (2018) A first look at browser-based crypto-jacking. In: 2018 IEEE European symposium on security and privacy workshops (EuroS&PW), pp 58–66. <https://ieeexplore.ieee.org/document/8406561>

# Real-Time Mood-Based Music Auto-Play System from Facial Expressions



Raunaq Singh, Vipin Singh, Pooja Verma, G.V. Eswara Rao,  
and Rajitha Bakthula

**Abstract** Human emotions are very much connected to the music what they want to play and reflects their current state of mind. Current music play systems are manual, i.e., they needs a manual input play-list creation and play them. Due to wide improvements in machine learning techniques, people are trying to automate all most all the tasks like: auto search based on voice tags, automated driving cars, automated cleaning systems, etc. Thus, this paper tries to use these machine learning techniques to read/detect the human behavior, i.e., emotion through camera and play audio songs automatically to based on this behavior(Only Hindi songs are considered for the study). It can reduce the manual efforts. Here, first the audio Hindi songs are classified and clustered into three groups: Happy, Sad, and Excited using machine learning techniques. Later the human face emotion is recognized using SVM and songs are played automatically.

**Keywords** Temporal and spatial audio feature · Facial recognition · Motion detection · Audio summarization · Precision · Recall

---

These authors contributed equally to this work.

R. Singh · V. Singh · P. Verma · G.V. E. Rao (✉) · R. Bakthula  
Computer Science and Engineering Department, Motilal Nehru National Institute of Technology,  
Teliarganj, Prayagraj 211004, Uttar Pradesh, India  
e-mail: [eswar nec01@gmail.com](mailto:eswar nec01@gmail.com)

P. Verma  
e-mail: [Poojaverma@mnnit.ac.in](mailto:Poojaverma@mnnit.ac.in)

R. Bakthula  
e-mail: [rajitha@mnnit.ac.in](mailto:rajitha@mnnit.ac.in)

## 1 Introduction

Music is often considered the “language of emotion”, and has been used widely to express a persons mental state. There will be five musical components in any music: melody, harmony, interval, rhythm, and pitch. These play a vital role in determining the psychological and physiological functions of a human. When people are happy they listen to joyous songs with higher pitch and tempo, whereas when they are sad they prefer listening to slow, low pitch songs. It has also been found by many medical researchers/scientists that when people listen to various musics their blood pressure varies according to the music, rate of heart beat also varies, and etc. Owing to this analysis many health experts use music as a therapy to cure ailments. It has been found by many theorists and researchers that non-verbal information prevails over words in human communication. Indian music has huge collection of songs including various languages like: Hindi, Marathi, Telugu, Tamil, Bhojpuri, and etc. Each language has their own vocal challenges in distinguishing the emotion of the song, i.e., sad, happy, excited etc. This paper presents study on only Hindi songs classification. In literature few researchers tried to classify the Hindi songs based on mood. Every song has basic information like genre, album name, year of album, artist name, and etc. are present as a part of music clip in the meta-data. But, this information may not play any role in designing or developing the automated music play systems (as they cannot determine the emotion of the song). Also, not many such mood-based music play-list generation applications exist in literature. In developing a music recommendation system, there are several factors that can contribute to bad music selection and reducing its efficiency. One of the major problem persist in most of the common music playing applications is it lacks the ability to acquire context of the user. So, the major factor to be considered is to look into the context of the activity that the user is doing/performing while listening to music. If the user is working on some task, they need a music track that does not take their concentration away, which is the antithesis of a good music recommendation system. Before designing any recommender system one should consider some important features like: what emotion is the user carrying?, what are his/her most viewed songs, what type of songs he prefers, and etc. Genre is one such parameter which judges the emotion of the song, it is the pattern or style followed in the music beat, instruments and rhythm. This can be a great factor while grouping or clustering the similar songs. Mental Exhaustion can also be recovered via music. It acts as a great tool in medical field to recover patients. Humans listen to variety of music depending on their mood and time in-order to relax. When they are stressed or tired they try to listen to music to recover from daunting state. A study was carried out using Contextual Sensor Data on people working on complicated tasks. Here music was played to relax them from tension and stress and its produced good results in terms of their energy level while working. Peoples mood can also be boosted by energetic or joyful songs. The static music recommendation system might fail to track the impact of emotion while working. The static systems just play the music files as per the folder not by the mood of the person. Thus, people stress might or might not get reduced by them

(static systems). This might effect their work performance as well as health. Hence a automated system might be used as a source to relieve pressure and stress by playing music as per the mood or emotion of the person. It can empower the persons ability and anxiety while working. These auto systems creation is challenging since a wrong selection of song or music (genre) might become more worser than the silence. A good recommendation system should tackle such situations and boost-up the users. Automated mood-based play-list generation has a large number of applications such as: Personalized Music Play System and Music Therapy. Music industry can also be benefited from this to identity the human reactions/likings toward the songs. The industry can then maximize its profits by determining the mood of the top selling songs. This paper, focuses on developing automated mood-based music play-list generation system (via songs classification based on audio features). The proposed model comprises two phases one for detection the face and identifying its emotion using the landmarks. Secondly a music type classification using the features extracted manually on song audio file. Then these two are composed further for auto-playing the music on dynamic detection of facial emotion. The remaining paper is organized as follows: Sect. 2 discusses the related work. Proposed model is detailed in Sect. 3. Experimental results and Performance analysis are presented in Sect. 4. Finally, Sect. 5 presents the conclusion.

## 2 Related Work

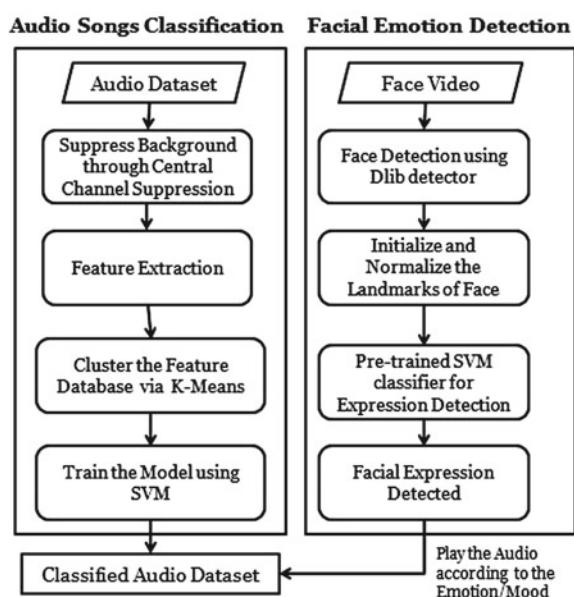
Several efforts have been made for mood-based music classification. One such work is given in [1], where mood classification of songs is done on the basis of lyrics. The song lyrics are analyzed and classified on the basis of occurrence of certain words which reflect sentiment. In their implementation, they have used features that fall into broadly two categories, Sentiment lexicon features and Text stylistic features. The paper [2] has performed three tasks: Recognition of user mood, tag the Hindi songs based on mood and finally play-list creation. They considered several moods such as Sad, Happy, Excited, Trance, and Romantic. Play-list creation module uses two modules outputs and matches them with the predefined mood of the user. Hampiholi [3] proposed an approach to classify the music samples of audio files using Genre Information (Present as part of meta-data) and other audio features. Ehshan et al. [4] designed an automated system using programming languages MySQL for data storage, Php, and JavaScript for designing the front end, clmtracker library has been used for recognizing the emotion. Karanam et al. [5] has studied the relationships between people and their health conditions on the basis of his/her interest in media files. This study was carried out to analyze the impact of health problems w.r.t media. Speaker recognition is performed by few authors such as [6–8] wherein they used temporal and facial feature were been used for speaker recognition and violence detection. Facial expressions and vocal data are used for emotion detection by Metallinou et al. [9] for analyzing the effectiveness of their proposed approach. Here, emotion is detected by marking some points on the facial image and trained them using GMM.

A song classification approach was proposed by Chen et al. [10] by splitting the given song vocal data into two parts and finally features are extracted on them. A hybrid approach combining learning-based approaches and keyword spotting was proposed by Amelia et al. [11]. A standard emotion detection approach was used for emotion classification into the classes of: disgust, fear, angry, sadness, happiness, and surprise. For learning the model they used support vector machine, Naive Bayes Multinomial model, Logistic regression model. A probability-based model was proposed by Kim et al. [12] for creating the recommender systems. Various researchers have proposed various techniques till date such as [13–23]. Emotisphere [24], a sensor-based interactive musical device was designed by Galen Chuang, this device plays the music as per the current emotional state of the user. A galvanic skin sensor and pulse sensor are used to recognize the emotion of the person(by placing the physiological marks on facial image). The user places his/her hand on the sensor for process initialization and the system automatically extracts the emotion and plays the music as per the mood.

### 3 Proposed Methodology

Proposed Model works in two phases: Phase-1: Automated Mood-Based Play-list Generation, Phase-2: Automated Facial Emotion Recognition. Overview of the proposed approach is shown in Fig. 1.

**Fig. 1** Overview of the proposed approach



### 3.1 Phase-1: Automated Mood-Based Play-List Generation

Implementation of automated play-list generation consists two sub tasks: Temporal and Spatial Audio Features Extraction, Mood-based feature classification and play-list generation.

#### 3.1.1 Temporal and Spatial Audio Features Extraction

This task extracts the audio features from songs dataset (Songs are in .wav format). This dataset is a collection of three different types of songs such as: happy, sad, and excited. Keeping in mind the varying nature of music in Bollywood songs, the song files have been preprocessed using Central Channel Extractor in Adobe Audition, to suppress the background music and enhance the vocals. jAudio feature extractor is used for extracting the features from audio clips. Each of the clips are divided into overlapping (of size 0.5) 32 milliseconds frames of size. Three types of feature categories are considered for the study such as: Intensity, Timbre, and Rhythm.

- **Timbre Features:** Timbre features are used to determine the sentiment of the emotion, i.e., the emotion is positive or negative. Features selected are: Strongest Frequency Via Zero Crossing, Roll off Point, Compactness, Strongest Frequency Via Spectral Centroid, Peak-based Spectral Smoothness, Flux, Strongest Frequency Via FFT Maximum.
- **Intensity Features:** Intensity features are used to identify the strength of the emotion. Features selected are Fraction of Low Energy Windows and RMS.
- **Rhythm Features:** Rhythm features are also to indicate whether the emotion is negative or positive. Usually fast beat songs tend to be happier than the slow beat songs. A total of 20 audio features have been extracted initially for each song in the dataset, and been used for classification. Later a feature reduction algorithm 20 is applied to remove unwanted features in-order to improve accuracy. Finally 15 features are selected for audio classification such as follows: (1) Spectral Flux Overall Standard Deviation (2) Strongest Beat Overall Standard Deviation (3) Beat Sum Overall Standard Deviation (4) Strongest Frequency via Spectral Centroid Overall Standard Deviation and Overall Average (5) Spectral Rolloff Point Overall Standard Deviation (6) Fraction of Low Energy Windows Overall Standard Deviation (7) Strongest Frequency via FFT Maximum Overall Standard Deviation and Overall Average (8) Compactness Overall Standard Deviation (9) Strongest Frequency via Zero Crossings Overall Standard Deviation and Overall Average (10) Spectral Rolloff Overall Average (11) Peak Based Spectral Smoothness Overall Standard Deviation and Overall Average (12) Root Mean Square Overall Average (13) Beat Sum Overall Average (14) Root Mean Square Overall Standard Deviation (15) Spectral Flux Overall Average For all these features overall average is calculated. These features of each video are stored in ARFF format for further processing.

### 3.1.2 Mood-Based Feature Classification and Play-List Generation

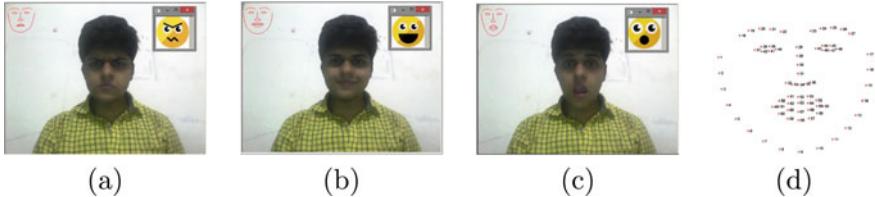
This task uses the Weka machine learning library [25–27] to classify and cluster the songs into their corresponding mood. Proposed method uses SMO algorithm for feature classification by splitting the features into training and testing sets. The trained model is used to classify the test audio features as Happy, Sad, and Excited classes, respectively.

### 3.1.3 Data Clustering

The K-means clustering algorithm is used to cluster the songs into the three mood categories. Here no. of clusters are chosen as 3 (as the categories considered are only 3). After the songs features are clustered, here each cluster is analyzed and song names are extracted from the path attribute present in the dataset. Finally the clustered features are used to generate the play-list. For clustering the experiments were conducted using K-means, PSO, and other algorithms while experiments and found that K-means is performing better for the considered dataset samples so we have used K-means as method for grouping the similar features.

## 3.2 Phase-2: Automated Facial Emotion Recognition

This Phase read the image of the user from camera and extracts the features from it and classifies using SVM. First task of this phase is to detect the face region from the image [28–31]. To analyze features of the face of a person in the image, it is very important to correctly extract the faces from a set of frames in the camera video clip. Here Dlib detector is used for the same, it makes use of a face detector, which detects faces using Histogram of Oriented Gradients (HOG) feature combined with a linear classifier and sliding window detection scheme. Once the face region is extracted from the frame it is scaled to extract the facial features. The most important process in the extraction of facial features is the landmark extraction localization. Face landmarks refer to the points on a persons face which uniquely define's its features. These landmarks clearly states an outline of the persons face, i.e., their eyes, lips, eyebrows, and nose etc. Dlibs landmark extractor detects 68 facial landmarks as shown in Fig. 2d. The pre-trained Dlib detector is developed using regression trees trained on iBug 300-W face landmark dataset. The landmarks are detected around the boundary of the face, but the distance between them might vary from image to image and user to user depending on the users face structure and their distance from the camera. Thus, using these landmarks for facial expression classification can cause a discrepancy in the experimental results. This limitation however, can be removed by performing normalization and scaling on the landmarks before computing the distance metrics. In this paper the normalized landmarks are detected on the normalized image of the detected face of size 100 pixels. Later Multiclass LibSVM



**Fig. 2** Facial emotion recognition samples. **a** The emotion detected as sad. **b** Emotion detected as Happy. **c** Emotion detection as excited. **d** Sample image of landmarks considered in the proposed method

tool is used to classify the emotion, i.e., Happy, Sad, and Excited. To classify the expressions, distance is used as a metric. The SVM is trained using the calculated distances between essential landmarks. Figure 2a–c shows some sample results on testing new samples on the trained SVM classifier.

In-order to prove the accuracy of the Facial Emotion Recognition, in this paper the Taiwanese Face Expression Image Database (TFEID) dataset [32] is used for experiments. This dataset is trained and tested using both SVM and Rule-based classifier.

## 4 Result Analysis

Experiments of audio songs classification were carried out on 75 audio songs consisting 25 songs of each of the categories: Happy, Sad, and Excited. All songs are in .wav format. Thus once trained the feature dataset will be of size 75 instances with 15 feature columns. Mood-based song classification has been done using Weka tool using SMO classification algorithm. The dataset is split into training and test sets. The model was trained using the training set and later tested on the test set and their accuracies were calculated as per Eqs. (1–5) (Table 1).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{TP Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

$$\text{FP Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

**Table 1** Comparison of proposed model with literature approaches

Approach	Method used	Accuracy (%)
Shashikumar et al. [33]	Labeled dataset with SVM	66
Lahoti et al. [34]	Mood statistics with Pi Chart and Bar Graph and CNN for classification	97
Naji et al. [35]	SVM	88.78
Naji et al. [36]	Bio & ECG signals and SVM	87.05
Sushmita et al. [37]	PCA and Euclidean Distance	93.75
Sana et al. [38]	Viola-Jones algorithm, Haar like features and CNN	95.89
Madhuri et al. [39]	CNN based sequential model	95
Proposed	Face landmarks and SVM	97.5

**Table 2** Comparison of detailed accuracy per class

Class	TP rate	FP rate	Precision	Recall
Happy	0.800	0.100	0.800	0.800
Sad	1.000	1.000	1.000	1.000
Excited	0.800	0.100	0.800	0.800

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

where, TP = No. of True Positives, FP = No. of False Positives, TN = No. of True Negatives and FN = No. of True Negatives.

Table 2 shows the detailed accuracy per class for 80–20% split in the dataset. Here for training the model the dataset is divided into 80% for training (60% for training and 20% for validation) and 20% for testing. Here we have used cross-validation while training. Table 3 shows the confusion matrix for songs classification, it can be observed from Table 3 that from a total of 15 test instances only 13 have been correctly classified and 2 have been incorrectly classified. Table 4 shows the confusion matrix for the K-means clustering results, it can be observed that the accuracy of the clustering is quite high, and only one instance out of 75 has been wrongly clustered, thus leading to 98.6% accuracy. Table 5 presents the comparison of emotions detected using rule-based classifier and SVM classifier, respectively. It can be observed from this table that SVM has performed better in classifying the emotions. Table 1 shows the comparison between proposed approach and literature approaches. It can be found from this table that the approaches which used handcrafted, CNN, PCA, SVM, and etc. for feature extraction and classification had less accuracy. Figure 3a, b depicts the comparison among the proposed approach using both the classifiers and the literature methods.

**Table 3** Confusion matrix for classification

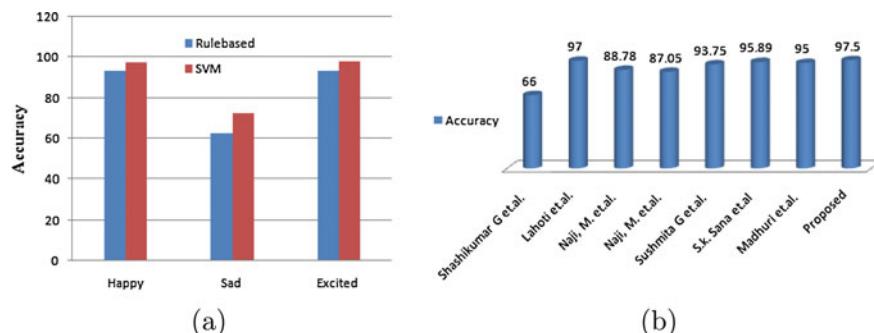
Emotion	Happy	Sad	Excited
Happy	4	0	1
Sad	0	5	0
Excited	1	0	4

**Table 4** Confusion matrix for simple K-means cluster

Emotion	Happy	Sad	Excited
Happy	25	0	0
Sad	0	25	0
Excited	1	0	24
Total	26	25	24

**Table 5** Comparison of facial expression classification

Emotion	Rule-based (%)	SVM (%)
Happy	93.33	97.5
Sad	62.5	72.5
Excited	93.57	98

**Fig. 3** **a** Comparison between the rulebased classifier and SVM classifier on the proposed feature extractor. **b** Comparing the accuracy between the proposed approach and literature methods

## 5 Conclusion

This paper is aimed to play the hindi songs based on the mood of the user. Here, firstly the hindi songs are clustered on the basis of emotion of the song such as Happy, Sad, and Excited. Secondly the facial emotion is tracked either from web camera of the systems and based on the mood recognized the songs are played. For song classification: Temporal Features & Spatial features are extracted then these

features are classified using SMO algorithm and finally audio songs are clustered. For Facial Emotion recognition: Dlib detector is used for initializing the landmarks then these landmarks values are trained using the SVM for Emotion recognition. Lastly, once the facial emotion is recognized the songs are played automatically to set the mood of the user.

## References

1. Fu Z, Lu G, Ting KM, Zhang D (2010) A survey of audio based music classification and annotation. *IEEE Trans Multimedia* 13(2):303–319
2. Liu CC, Yang YH, Wu, PH, Chen H (2006) Detecting and classifying emotion in popular music. In: 9th joint international conference on information sciences (JCIS-06). Atlantis Press. <https://doi.org/10.2991/jcis.2006.325>
3. Hampiholi V (2012) A method for music classification based on perceived mood detection for Indian bollywood music. In: Proceedings of world academy of science, engineering and technology, no 72, p 507. World Academy of Science, Engineering and Technology (WASET)
4. Ehshan A, Dider M, Haque N, Sagor M (2018) Emotion based song recommendation. Daffodil International University, PhD diss
5. Karanam K, Edward Martucci W, Kopikis A, Zohar D, Kuchekar NN, Marsh AB, Sian SA, Krishnan V (2018) Systems and techniques for identifying and exploiting relationships between media consumption and health. U.S. Patent Application 15/721,763, filed August 23
6. Nemani P, Krishna GS, Ramisetty N, Sai BD, Kumar S (2022) Deep learning based holistic speaker independent visual speech recognition. In: IEEE transactions on artificial intelligence. <https://doi.org/10.1109/TAI.2022.3220190>
7. Krishna GS, Supriya K, Vardhan J (2022) Vision transformers and YoloV5 based driver drowsiness detection framework. arXiv preprint [arXiv:2209.01401](https://arxiv.org/abs/2209.01401)
8. Singh S, Dewangan S, Krishna GS, Tyagi V, Reddy S (2022) Video vision transformers for violence detection. arXiv preprint [arXiv:2209.03561](https://arxiv.org/abs/2209.03561)
9. Metallinou A, Lee S, Narayanan S (2008) Audio-visual emotion recognition using gaussian mixture models for face and voice. In: 2008 Tenth IEEE international symposium on multimedia, pp 250–257
10. Chen CH, Lu PT, Chen OTC (2010) Classification of four affective modes in online songs and speeches. In: The 19th annual wireless and optical communications conference (WOCC 2010), pp 1–4
11. Amelia W, Maulidevi NU (2016) Dominant emotion recognition in short story using keyword spotting technique and learning-based method. In: 2016 international conference on advanced informatics: concepts theory and application (ICAICTA), pp 1–6
12. Kim J, Lee S, Yoo W (2013) Implementation and analysis of mood-based music recommendation system. In: 2013 15th international conference on advanced communications technology (ICACT), pp 740–743
13. Li J, Tang J, Liu H (2017) Reconstruction-based unsupervised feature selection: an embedded approach. IJCAI. 2017. <http://www.ijcai.org/proceedings/2017/0300>
14. Sen A, Larson M (2015) From sensors to songs: a learning-free novel music recommendation system using contextual sensor data. In: LocalRec@RecSys, Sep 19, pp 40–43
15. Boothby DM, Robbins SJ (2011) The effects of music listening and art production on negative mood: a randomized, controlled trial. *The Arts Psychother* 138(3):204–8
16. Houston Behavioral, The Top 3 Causes of Stress in Today's World <http://www.houstonbehavioralhealth.com/blog/top-3-causes-of-stress>. Accessed 11 Apr 2019
17. Wu PY, Huang ML, Lee WP, Wang C, Shih WM (2017) Effects of music listening on anxiety and physiological responses in patients undergoing awake craniotomy. *Complement Ther Med* 32:56–60

18. Raju MK, Shafiulilah SK (2015) An enhanced risk prediction system for cardiovascular disease in India using fuzzy classification. *Int J Comput Appl* 975:8887
19. Majma N, Babamir SM, Monadjemi A (2017) Runtime verification of pacemaker functionality using hierarchical fuzzy colored Petri-Nets. *J Med Syst* 41(2):27
20. Natividad MC, Gerardo BD, Medina RP (2019) A fuzzy-based career recommender system for senior high school students in K to 12 education 482(1):012025
21. Li F, Islam SR, Kwak D, Khan P, Ullah N, Yoo SJ, Kwak KS (2018) Type-2 fuzzy ontology aided recommendation systems for IoT based healthcare. *Comput Commun* 119:138–55
22. Chaturvedi V, Kaur AB, Varshney V, Garg A, Chhabra GS, Kumar M (2022) Music mood and human emotion recognition based on physiological signals: a systematic review. *Multimedia Syst* 28(1):21–44
23. Krishnaiah A, Divakarachari PB (2021) Automatic music mood classification using multi-class support vector machine based on hybrid spectral features. *Int J Intell Eng Syst* 14.5: 102–111. <https://doi.org/10.22266/ijies2021.1031.10>
24. Chuang G (2015) EmotiSphere: from emotion to music. In: 2015 TEI15 proceedings of the ninth international conference on tangible embedded embodied interaction, pp 599–602
25. Gorunescu F (2011) Data mining: concepts, models and techniques, vol 12. Springer Science & Business Media
26. Pulli K, Baksheev A, Korniyakov K, Eruhimov V (2012) Real-time computer vision with OpenCV. *Communications ACM* 55(6):61–69
27. King DE (2009) Dlib-ml: a machine learning toolkit. *J Mach Learn Res* 10(Jul): 1755–1758
28. Hsu RL, Abdel-Mottaleb M, Jain AK (2002) Face detection in color images. *IEEE Trans Pattern Anal Mach Intell* 24(5):696–706
29. Wu W, Zhang Y, Li C, Qian C, Loy CC (2018) Reenactgan: Learning to reenact faces via boundary transfer. In: Proceedings of the European conference on computer vision (ECCV), pp 603–619
30. Ko B (2018) A brief review of facial emotion recognition based on visual information. *Sensors* 18(2):401
31. Dangi K, Kushwaha MS, Bakthula R (2019) An intelligent traffic light control system based on density of traffic. In: Emerging technology in modelling and graphics, pp 741–752. Springer, Singapore
32. Li-Fen Chen and Yu-Shiuan Yen. (2007). Taiwanese Facial Expression Image Database [<http://bml.ym.edu.tw/download/html>]. Brain Mapping Laboratory, Institute of Brain Science, National YangMing University, Taipei, Taiwan
33. Totad SG, Patil SN, Bharamagoudar GR (2022) Emotion based music classification. *J Next Gener Technol* 2(1). ISSN: 2583-021X
34. Lahoti M, Gajam S, Kasat A, Raul N (2022) Music recommendation system based on facial mood detection. In: 2022 third international conference on intelligent computing instrumentation and control technologies (ICICICT). IEEE, pp 284–289
35. Naji M, Firoozabadi M, Azadfallah P (2013) Classification of music-induced emotions based on information fusion of forehead biosignals and electrocardiogram. *Cognit Comput* 6(2):241–252
36. Naji M, Firoozabadi M, Azadfallah P (2015) Emotion classification during music listening from forehead biosignals. *Signal Image Video Process* 9(6):1365–1375. <https://doi.org/10.1007/s11760-013-0591-6>
37. Kamble SG, Kulkarni AH (2016) Facial expression based music player. In: 2016 international conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 561–566
38. Sana SK, Sruthi G, Suresh D, Rajesh G, Subba Reddy GV (2022) Facial emotion recognition based music system using convolutional neural networks. *Mater Today: Proc* 62:4699–4706
39. Athavle M, Mudale D, Shrivastav U, Gupta M (2021) Music recommendation based on face emotion recognition. *J Inf Electrical and Electr Eng (JIEEE)* 2(2):1–11

# A Wrapper-based Feature Selection Approach Using Particle Swarm Optimization for Software Fault Prediction



Hritik Shah and Himansu Das

**Abstract** The main aim of feature selection (FS) is to determine and select the optimal number of features from a big dataset to help us find more accurate predictions with high efficiency and less computation power. Particle Swarm Optimization (PSO) is a wrapper-based feature selection algorithm which is inspired by flocking of bird and schooling of fishes. The primary goal here is to put forward a way for earlier software fault prediction using PSO-based feature selection. Through an early detection of a software fault, we not only speed up the future software development process but also save a lot of time and resources. In this article, we have also attempted to show a brief comparison of accuracy between different classifiers namely—K-nearest neighbors, Naïve Bayes and decision tree when used with PSO. Our study shows a promising increase in accuracy when PSO with all three above-mentioned classifiers was used for various datasets.

**Keywords** Particle swarm optimization · Feature selection · Software fault prediction · Classifiers

## 1 Introduction

Software fault prediction (SFP) [1] model helps to identify software faults in early stage. SFP uses an analytical review of previous software metrics and fault data to detect software faults. The main aim of SFP is to identify faults so that a huge amount of money, time and human effort can be saved which was to be used to correct them. In addition to that the quality of software can also be increased and also efficiency

---

H. Shah · H. Das (✉)

School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha 751024, India

e-mail: [das.himansu2007@gmail.com](mailto:das.himansu2007@gmail.com)

H. Shah

e-mail: [1905095@kiit.ac.in](mailto:1905095@kiit.ac.in)

can be improved. If SFP was not there, then software fault would have been detected later in process which would cause engineers to re-evaluate and re-design the entire process which would be very troublesome hence SFP is very useful and necessary.

As the data flow is increasing, massive datasets with complex and significantly more features are being stored. From this huge pool of data, there are many numbers of those features which are not required for classification [2] process and did not play any role in affecting the output prediction. This created a need for developing certain techniques to extract only important features from large datasets and feeding those features to a machine learning [3] algorithm to get an optimal solution with greater efficiency. This concept helped to discover numerous feature selection [4, 5] approaches.

The selection technique is being used for many machine learning algorithms as a preprocessing and data-cleaning tool. Its main aim is to take away redundant attributes thereby decreasing the outliers in our classification and giving us more accurate and faster results. FS can be categorized into—filter, wrapper and embedded. Usually, all the above-mentioned feature selection categories like filter, wrapper and embedded are generally slow and require some kind of optimization for faster performance; hence, we have used PSO for the optimization purpose.

PSO is a virtual simulation of the simplified social system which is greatly influenced by birds flocking [6]. An individual bird's observable range is very limited, but a swarm of birds through communication can be aware of a larger area and find its food. Similarly in PSO, a random initialization of particle is performed and its solution traverse in search space to find a more optimal and better solution in every iteration by regularly updating its velocity and position [7] using parameter like gbest and pbest.

In this paper, we put forward PSO, a wrapper-based FS algorithm which helps us filter out the most important subset features require for our classification. This wrapper-based FS method is based on ML algorithm which evaluate many models using techniques that change predictors to find the best model solution. We used fitness values (i.e., error) as the evaluation criteria which means that during implementation in coding we calculated the fitness by first calculating the error (amount of mismatched in training and testing data), and accuracy was calculated as it is inversely proportional to error. In this method, we used PSO with a KNN classifier, Naïve Bayes classifier and decision tree classifier [8], respectively, and analyzed the results of them together to arrive at a conclusion.

The below paper divided as Sect. 2 contains related works and literature review, Sect. 3 and its subdivision give us details about FS PSO, Sect. 4 discusses result analysis, and finally Sect. 5 concludes the paper.

## 2 Related Work and Literature Review

Feature selection is to identify and select the optimal number of features from a big dataset to help us find more accurate predictions with high efficiency and less computation power. This section provides us the information of previous work done by different people on feature selection approaches using PSO and software fault predictions.

Although PSO, as an algorithm, has its fast convergence, which compares favorably with many global optimization algorithms like genetic algorithms (GA), simulated annealing (SA) and other global optimization algorithms, one of its limitations is that the optimum local searchability of the PSO algorithm is weak and even the trend of creating a new variant PSO in case of swarm robotic projects is also getting alarming.

Can et al. [9] proposed a new model for detecting software failures called P-SVM Model using PSO and SVM. This model took benefit of nonlinear computing ability of SVM and optimization of parameter of PSO. It used PSO algorithm to calculate the SVM's best parameter and adopted optimized SVM for software fault prediction (SFP).

Zheng et al. [10] introduced a method called FS-JMIE where JMIE (a metric) searches feature subset performance. He then applies Binary Particle Swarm Optimization (BPSO) for subset searching.

Shehu et al. [11] proposed a time-based PSO variant. They did this by establishing a time constant in update function of velocity through which they avoided premature convergence, particularly in the dataset of emotion frame of videos. This PSO variant is compared with the two standard PSO algorithm. This helped them categorize emotion from (CK + ) dataset.

Tan et al. [12] defined K-nearest neighbor (KNN) as a nonparametric algorithm that assumes similar attributes between new and assumed cases and then allocates the new case into the group that is most matching to the available groups.

Mitchell et al. [13, 14] considered Naive Bayes as a kind of probabilistic classifier. It consists of a group of algorithms where all the algorithms have a common goal which is that every pair of attributes that is being classified must be self-reliant.

Yang et al. [15] described the goal of this algorithm is to create the decision boundaries (hyperplane) which can differentiate space of n-dimension into respective classes making the algorithm to easily place the new data points in the right group/categories. SVM chooses the extreme points (the point which is closer to the opponent class) which helps in implementing the hyperplane. The above-mentioned edge cases are support vectors. Thus, this algorithm is called Support Vector Machine.

### 3 Feature Selection

As already discussed, feature selection (FS) is used to select the required features and reduce irrelevant features, thus decreasing the effects related to the curse of dimensionality [16] hence increasing efficiency. It can be categorized into:

- Filter Method—It uses a statistical computation to compute the relevancy of the predictors which are outside of the predictive models and retain only the predictors that pass certain criteria.
- Wrapper method—It evaluates several models using technique which adds or remove predictors to get the required combination that will maximize the performance of model.
- Embedded method—Embedded methods are models in which feature selection method happens naturally in the process of the model fitting. Put simply, this method integrates the feature selection algorithm as part of the machine learning algorithm. The most typical embedded technique is a tree-based algorithm, which includes a decision tree and a random forest.

#### 3.1 Particle Swarm Optimization

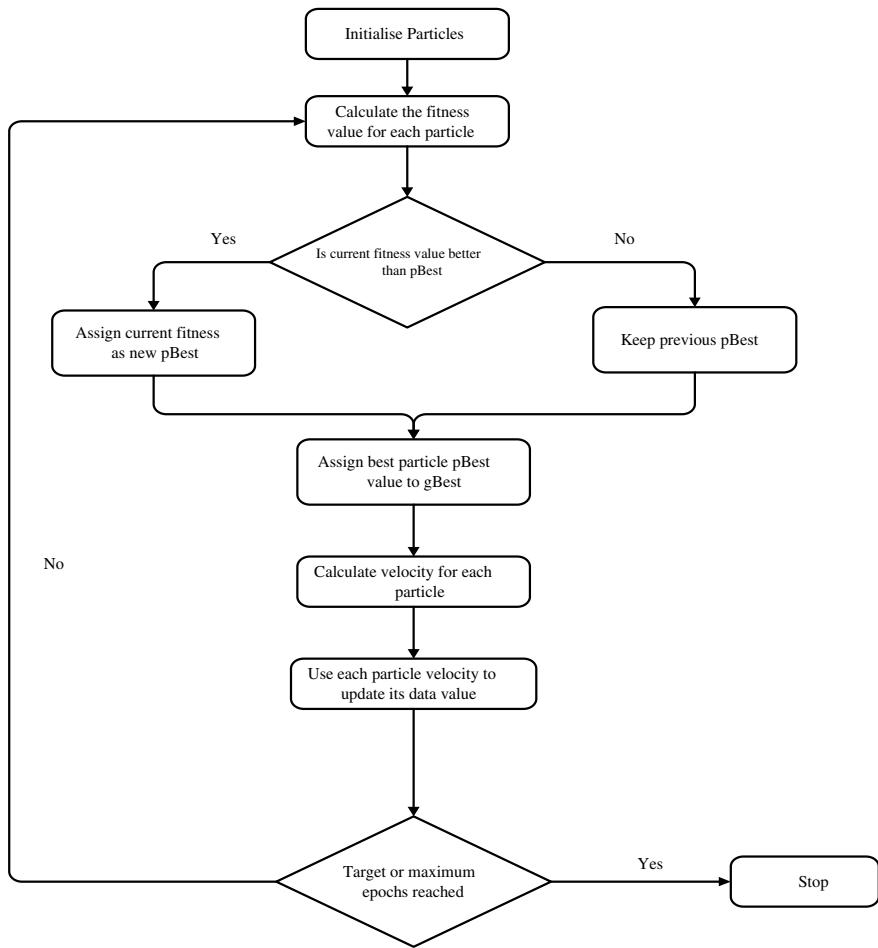
PSO is a kind of population-based meta-heuristic algorithm made by looking social behavior like school of fish and flock of birds [6]. The main principle of PSO is that information is being optimized not just by personal experience rather through social interaction in the population [17]. In PSO, every individual particle is a point in the search space of d-dimension with its own respective memories that store its best experience with the optimal event that was come across by the swarm as team to get optimal answer/solution.

In the search space vector, the preliminary particle's population is distributed at random. Each particle's position is denoted as a vector where  $d \in D$  represents dth dimension in our search space [7]. Particles happens to move in search space with position  $x_i$  and with velocity  $v_i$ . In course of their movement particles update their velocity as well as position according to their adjacent ones. Every particle remembers and stores its position where it had the best experience and this is called a particle's  $p_{best}$  (personal best). Out of all the  $p_{best}$  of each particle, the overall best is called  $g_{best}$  (group best). The swarm tries to find the best and the most optimal solution by updating their position and velocity for each particle after every iteration and this  $p_{best}$  and  $g_{best}$  has an important role in this as per following equation.

$$x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \quad (1)$$

$$v_{id}(t + 1) = w^* v_{id}(t) + c_1^* r_1^*(p_{id} - x_{id}(t)) + c_2^* r_2^*(p_{gd} - x_{id}(t)) \quad (2)$$

Equation (1) represents the updated position and Eq. (2) represents the updated velocity.  $p_{id}$  is the personal best and  $p_{gd}$  is the global best and  $w$  denotes inertia weight which manipulates the effect of last velocities on the present velocity.  $t$  is the  $t$ th iteration,  $x_{id}(t)$  represents of  $i$ th particle position of in  $t$ th iteration in  $d$ th dimension of search space,  $v_{id}(t)$  represents velocity of  $i$ th particle position of in  $t$ th iteration in  $d$ th dimension of search space.  $c_1$  and  $c_2$  are the learning constant,  $r_1$  and  $r_2$  are simple random values uniformly distributed in  $[0,1]$ . Figure 1 represents the flow of the feature selection using PSO algorithm.



**Fig. 1** Flowchart of the PSO algorithm

### 3.2 Binary Particle Swarm Optimization (BPSO)

In BPSO [18], the velocity parameter is still being changed and updated as it used to be in normal standard PSO. However, variables  $x_{id}$ ,  $p_{id}$  and  $p_{gd}$  can only be assigned the value 0 and 1. Hence by this approach, velocity would point to the particle's probability in the position vector to have the value 1. In BPSO, the current particle's position is shown as Eq. (3) based on the probability value  $T(v_t)$  which we got from Eq. (4).

$$x(t+1) = \begin{cases} 1, & \text{If } \text{rand} < S(v(t+1)) \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

where  $S(v(t))$  denotes the Sigmoid function as shown and rand is random number  $\in [0, 1]$ .

$$S(v(t)) = \frac{1}{1 + e^{-v(t)}} \quad (4)$$

### 3.3 Feature Selection Using Particle Swarm Optimization

At each step, every particle is getting updated by keeping two of its best values, i.e., personal best ( $p_{best}$ ) and global best ( $g_{best}$ ). Each individual particle keeps records of its position coordinate which is its best solution or fitness which has been achieved so far. This fitness value is  $p_{best}$ . Thus, when all particles behave as a swarm out of all the  $p_{best}$  of every particle, one best coordinate is selected and is called  $g_{best}$ . As all features is ultimately an attribute, so we assign either 0 or 1 as discussed in BPSO and we select only those features who is assigned 1 and neglect the one assigned 0. In this way, features are selected. The pseudo-implementation for the PSO using feature selection is:

1. Initialize 0 to all particles initial speed for 0th iteration
2. Initialize the starting particle position as  $x = x_{min} + \text{rand} [0, 1]$
3. Initialize  $p_{best}$  which for 0th iteration is same as initial position of particle's velocity.  $G_{best}$  is taken from  $p_{best}$  with maximum fitness.
4. Velocity value is taken from the addition of momentum and experience which is got from  $g_{best}$  and  $p_{best}$  which is given by Eq. (5).

$$\begin{aligned} VB = w^*V[i, j] + c1^*\text{random} \cdot \text{random}()^*(p_{best}[i, j] - X[i, j]) \\ + c2^*\text{random} \cdot \text{random}()^*(g_{best}[j] - X[i, j]) \end{aligned} \quad (5)$$

5. Update position and calculate fitness as represented by Eq. (6).

**Table 1** Datasets and it's description used for the algorithm

S. No.	Datasets	No. of instances	No. of features
1	CM1	327	38
2	JM1	7782	22
3	KC1	1183	22
4	KC3	194	40
5	MC1	1988	39
6	MC2	125	40
7	MW1	253	38
8	PC1	705	38
9	PC2	745	37
10	PC3	1077	38
11	PC4	1287	38
12	PC5	1711	39

$$X[i, j] = X[i, j] + V[i, j] \quad (6)$$

6. Compare the previous pbest and the output of the position update and the best one will be assigned to gbest.

## 4 Result Analysis

In this analysis, we have utilized 12 datasets to perform feature selection through the PSO algorithm applying three different classifiers to each dataset. Additionally, a graphical representation of the fitness values obtained through FS PSO for each classifier has been plotted, and a reference table has been provided to compare the accuracy of the classifiers with and without FS PSO.

### 4.1 Dataset Used

Here, we have used 12 open-source datasets from NASA Repository [19]. The details about the dataset is given in Table 1.

### 4.2 Experimental Condition

The experiment were performed in Jupyter Notebook 6.4.8 using Python version 3.9.11. The specification of hardware configuration is Intel Core i5-8250U processor

with 16 GB RAM and NVIDIA GeForce MX130. In the experiments, inertia weight wmax was set to 0.9 and wmin to 0.4. The acceleration factor c1 and c2 was set to 2. The size of the population was set to 10, and the maximum iteration was set to 200.

### **4.3 Experimental Results**

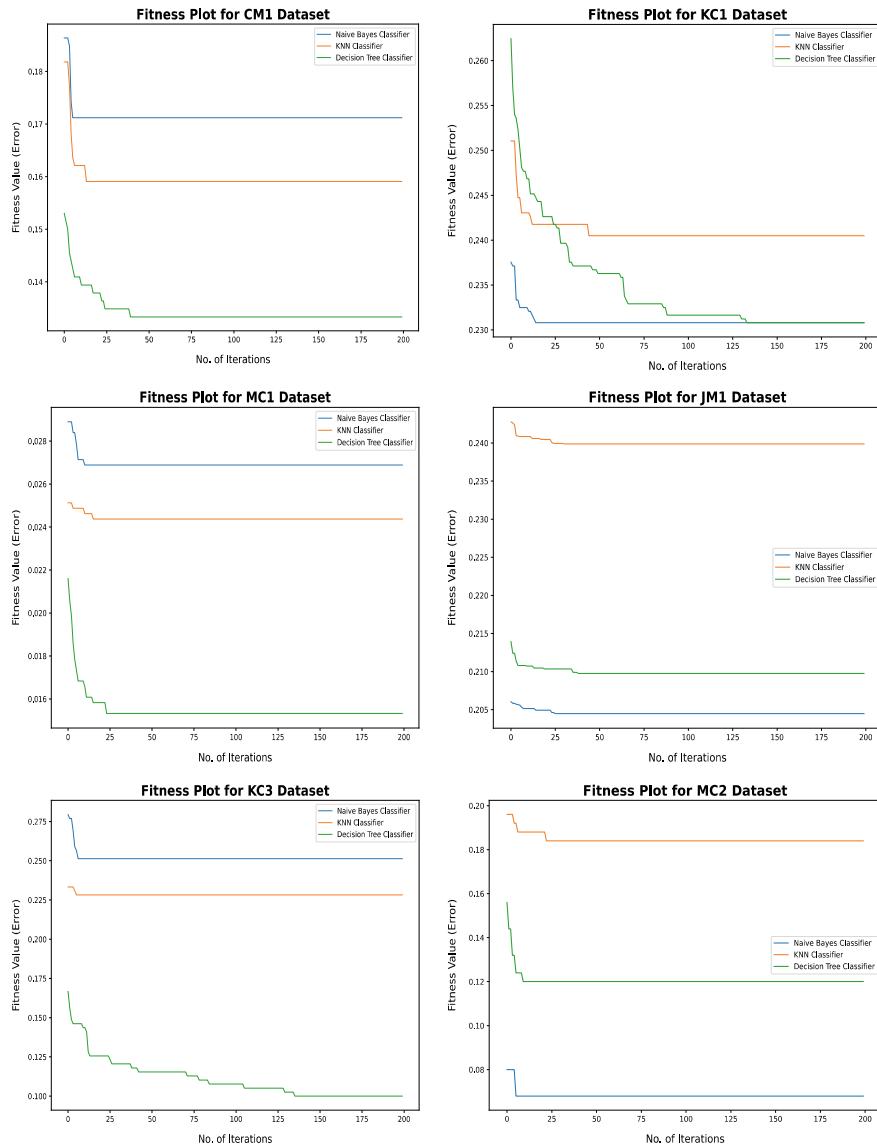
Experiments were performed on all the 12 datasets and their respective average fitness (error) plots for all the datasets after applying PSO algorithm using KNN, Naïve Bayes classifier and decision tree classifier are show below. Figure 2 represents fitness (error) plots for the datasets CM1, KC1, MC1, JM1, KC3 and MC2, and Fig. 3 represents fitness (error) plots for the datasets MW1, PC1, PC4, PC3, PC2 and PC5.

Table 2 contains the accuracy with and without FS PSO for all 12 datasets. Here, the accuracy as well as number of features selected are the average taken out for 10 observations. Floating point features selection number means average number of features that was selected after 10 observations and does not account for number of features selected for 1 observation. For a single observation, number of features selected will always be an integer.

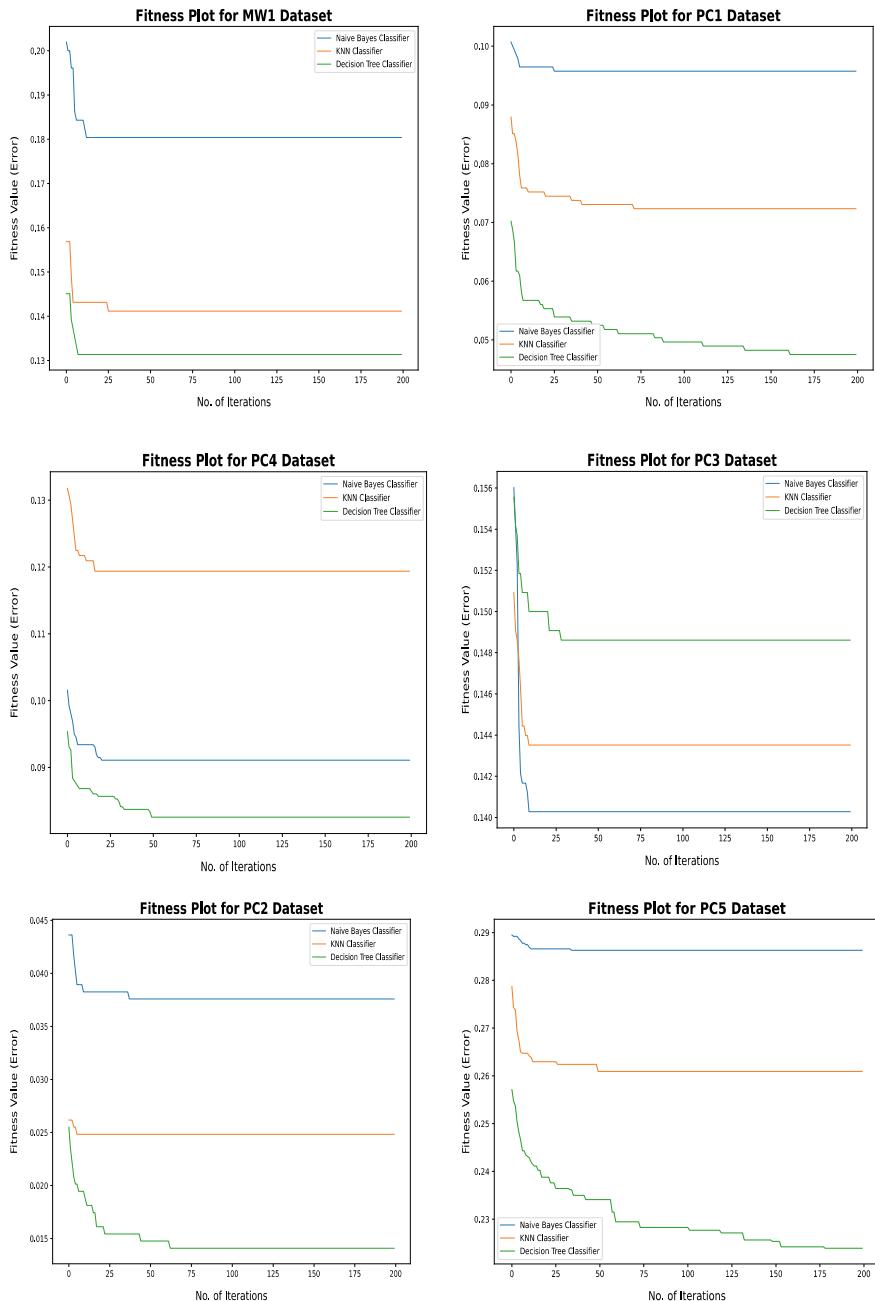
The experimental result of feature selection using PSO shows us that the accuracy of prediction has increased for all the three different classifiers. Comparing it with previous values in Table 2, we see that accuracy with FS PSO has increased for all. We can also see that the unnecessary feature was discarded and only useful one was selected for optimization.

## **5 Conclusion**

Here, we proposed feature selection model by the use of PSO to detect software prediction fault. Here, we used 12 datasets, and for each dataset, three different classifiers were used namely KNN, Naïve Bayes (NB) and decision tree (D-tree). The main aim/goal of this paper was to find the most optimal features in the dataset and the results of experiment clearly show us that the proposed feature selection using PSO increased the prediction accuracy. In addition, the prediction accuracy was increased for all the classifier in all datasets which is visible form Table 2. Also, the non-important features were removed. Hence, it can be concluded that the prediction accuracy was increased in all three classifiers by using feature selection using PSO. This algorithm has witnessed many enhancements in various aspects of the algorithm and as a result the related research has now reached an impressive state and more reviews also highlight the current issues and more open challenges haunting PSO, prompting scholars and researchers to conduct further research on both the theory and the application of the algorithm in the coming years.



**Fig. 2** Fitness (error) plots for the datasets CM1, KC1, MC1, JM1, KC3 and MC2



**Fig. 3** Fitness (error) plots for the datasets MW1, PC1, PC4, PC3, PC2 and PC5

**Table 2** Table containing accuracy with and without FS PSO for all 12 datasets

S. No.	Datasets	Classifiers used	Accuracy without FS	Accuracy with FS PSO	No of selected features
1	CM1	Naive Bayes	77.2727	81.96969	12.7
		KNN	75.7576	83.3333	12.7
		D-Tree	78.7878	81.9696	14.4
		Naive Bayes	78.998	79.1586	9.4
2	JM1	KNN	73.3461	75.061	8.4
		D-Tree	69.9422	71.811	12.4
		Naive Bayes	74.2616	76.1181	7.3
3	KC1	KNN	69.6203	74.641	6.7
		D-Tree	72.1519	73.122	11.2
		Naive Bayes	66.6667	71.538	15.5
4	KC3	KNN	74.3589	76.1538	18.4
		D-Tree	76.9231	84.615	17.5
		Naive Bayes	95.7286	96.3065	13.9
5	MC1	KNN	96.4824	97.5628	16.9
		D-Tree	96.482	97.286	11.6
		Naive Bayes	92.0	92.8	17.9
6	MC2	KNN	76.0	80.0	18.3
		D-Tree	64.0	76.0	17.9
		Naive Bayes	76.4706	78.8235	15.2
7	MW1	KNN	78.4314	84.7058	13.4
		D-Tree	80.3921	82.1568	14.2
		Naive Bayes	87.2340	89.6453	14.0
8	PC1	KNN	89.3617	90.8510	12.4
		D-Tree	88.6525	92.0567	22.0
		Naive Bayes	93.9597	95.8395	12.2
9	PC2	KNN	96.6443	97.4496	15.4
		D-Tree	95.3020	96.51006	16.5
		Naive Bayes	68.9815	80.8333	11.7
10	PC3	KNN	82.4074	84.7685	14.7
		D-Tree	78.7037	82.9166	15.0
		Naive Bayes	86.8217	89.1085	14.4
11	PC4	KNN	84.4961	86.6279	13.3
		D-Tree	89.5348	90.34883	14.5
		Naive Bayes	70.5539	70.8163	15.1
12	PC5	KNN	67.0554	71.982	12.6
		D-Tree	72.5948	73.5276	21.7

## References

1. Cagatay C, Banu D (2009) A systematic review of software fault prediction studies. *Expert Syst Appl* 36(4):7346–7354
2. Das H, Naik B, Behera HS (2020) An experimental analysis of machine learning classification algorithms on biomedical data. In: Proceedings of the 2nd international conference on communication, devices and computing: ICCDC 2019. Springer, Singapore. [https://doi.org/10.1007/978-981-15-0829-5\\_51](https://doi.org/10.1007/978-981-15-0829-5_51)
3. Das H, Naik B, Behera HS (2021) Optimal selection of features using artificial electric field algorithm for classification. *Arabian J Sci Eng* 46:9:8355–8369
4. Das H et al. (2020) Optimal selection of features using teaching-learning-based optimization algorithm for classification. In: Applied intelligent decision making in machine learning. CRC Press, pp 213–227
5. Das H, Naik B, Behera HS (2022) A Jaya algorithm based wrapper method for optimal feature selection in supervised classification. *J King Saud Univer-Comput Inform Sci* 34:6:3851–3863
6. Yuhui S, Russell C (1999) Empirical study of particle swarm optimization. In: Evolutionary computation, CEC 99. Proceedings of the 1999 congress on, vol 3. IEEE, pp 1945–1950
7. Himangshu B, Thakur J, Satyajit S, Nazrul H (2020) A feature selection method using PSO-MI, pp 1–2
8. Iliou T, Anagnostopoulos N (2009) Comparison of different classifiers for emotion recognition. In: 2009 13th Panhellenic conference on informatics, pp 102–106. <https://doi.org/10.1109/PCI.2009.7>
9. Can H, Jianchun X, Ruide X, Juelong L, Qiliang Y, Liqiang X (2013) A new model for software defect prediction using particle swarm optimization and support vector machine. In: 2013 25th Chinese control and decision conference (CCDC), pp 4106–4110. <https://doi.org/10.1109/CCDC.2013.6561670>
10. Zheng K, Wang X (2018) Feature selection method with joint maximal information entropy between features and class. *Pattern Recogn* 77:20–29
11. Shehu A, Browne W, Eisenbarth H (2021) Particle swarm optimization for feature selection in emotion categorization. In: 2021 IEEE congress on evolutionary computation (CEC), 2021, pp 752–759. <https://doi.org/10.1109/CEC45853.2021.9504986>
12. Tan S (2006) An effective refinement strategy for KNN text classifier. *Expert Syst Appl* 30(2):290–298
13. Mitchell M (2007) Machine learning. Burr Ridge, IL, McGraw Hill, vol 45(37). pp 870–877
14. Michie D, Spiegelhalter J, Taylor C (1994) Machine learning, neural and statistical classification, pp 1–298
15. Yang Y, Jianping L, Yimei Y (2015) The research of the fast SVM classifier method. In: 2015 12th international computer conference on wavelet active media technology and information processing (ICCWAMTIP), pp 121–124
16. Sharma N, Saroha K (2015) Study of dimension reduction methodologies in data mining. In: International conference on computing, communication and automation 2015, IEEE, pp 133–137
17. Xue B, Zhang M, Browne N (2013) Particle swarm optimization for feature selection in classification: a multi-objective approach. *IEEE Trans Cybernet* 43(6):1656–1671
18. Majdi M (Ed) (2018) Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In: Proceedings of the 2nd international conference on future networks and distributed systems, pp 1–9
19. Shepperd M, Song Q, Sun Z, Mair C (2013) Data quality: some comments on the nasa software defect datasets. *IEEE Trans Softw Eng* 39:1208–1215

# Sign Language to Sentence Interpreter Using Convolutional Neural Network in Real Time



S. Seetha, C. Christlin Shanuja, Esther Daniel, Saurabh Chandra, and Saurabh Raj

**Abstract** Sign languages are used by several people across the globe as their primary language. Sign language is a non-verbal language that people use to communicate with each other by using facial/body expressions, postures, and a set of gestures. It is based on visual orientation and is natural. It is primarily used as a means of communication with hearing-impaired or deaf individuals. Artificial intelligence (AI) has been utilized to break down communication barriers between the hearing and deaf communities, seeking to create automatic sign language generation and recognition systems as well as to “connect with the machines”. This article discusses about a desktop application that interprets finger spelling based on American Sign Language in real time using neural networks. In this method, the hand gesture is photographed using a camera, put through a filter that eliminates RGB background noise and utilizes a Gaussian filter to capture the edges of the hands, and then put through a classifier that determines the type of hand motion it belongs to. The character is then shown, and a sentence is then formed by adding both. The employed method has a 98.7% accuracy rate for the alphabet’s 26 letters.

**Keywords** Convolutional neural network · Computer vision · Hand gestures · Hand postures · Gesture classification · Machine Learning · Sign Language Recognition

---

S. Seetha (✉) · C. Christlin Shanuja · S. Chandra · S. Raj  
CMR Institute of Technology, Bengaluru 560037, India  
e-mail: [sitamismin19@gmail.com](mailto:sitamismin19@gmail.com)

C. Christlin Shanuja  
e-mail: [christlin.c@cmrit.ac.in](mailto:christlin.c@cmrit.ac.in)

S. Chandra  
e-mail: [sach18is@cmrit.ac.in](mailto:sach18is@cmrit.ac.in)

S. Raj  
e-mail: [sara18is@cmrit.ac.in](mailto:sara18is@cmrit.ac.in)

E. Daniel  
Karunya Institute of Technology and Science, Coimbatore 641114, India

## 1 Introduction

The vision-based Sign Language Translation (SLT) resolves the problem of linking the communication between deaf and normal people and also the machines. Instead of using acoustic sound patterns to communicate word meanings, sign language physically transmits sign patterns using hand forms, arm orientation and movements, facial expressions, and lip patterns. Around the world, there are numerous sign languages, each with its unique vocabulary and motions.

There are lot of research [6–13] has been done with respect to Sign Language Translation, and some examples are American Sign Language (ASL), British Sign Language (BSL), and so on. There are approximately 6000 gestures of ASL for common words, using fingers to communicate obscure words or proper nouns. For the past two decades, the field of Sign Language Recognition (SLR) has been the focus of intensive research. However, in the majority of research [14–20], SLR is seen as a naïve gesture recognition problem. SLR aims to identify a series of continuous signs while ignoring the rich grammatical and syntactic elements that distinguish sign language from spoken language. In contrast, the Sign Language Translation (SLT) problem is introduced. The problem of developing Sign Language Recognition ranges from data collection to building an effective model for image recognition. Using a camera can be inefficient because of the noise it produces, which demands extensive image preprocessing, and using a sensor-based technique can be expensive and impractical. Using machines that translate sign language into speech is one option, but it is not a practical one because of the high price and ongoing maintenance involved. Feature extraction and classification are the traditional methods for Sign Language Identification.

## 2 Related Works

The comparison of existing techniques used for Sign Language Identification using Machine Learning algorithms was discussed in this section. Kau et al. [1] describe a hand-free illustration of Taiwanese data language which is used to process the data by using the wireless system. The internal sensors built into gloves can distinguish between distinct hand motions by displaying the properties of the hand's posture, orientation, and motion as defined by Taiwanese Sign Language. The flex inner sensor helps to identify the hand gesture, the 'g' sensor helps to identify the size of the palm, and the gyroscope helps to capture the movement. It would be necessary to periodically examine input signals to determine whether the sign was legal or not. The legal gesture is communicated via the phone with connectivity like Bluetooth for discriminating and translating gestures. The accuracy of gesture recognition with the suggested architecture and algorithm is quite good. The concurrent design yields an accuracy of 94% for the result. A real-time sign language identifier tries to provide a quality communication for the disabled persons, such as hearing and deaf society.

Shahriar et al. [2] presented the goal of skin color conversion to recognize American Sign Language (ASL) using Machine Learning algorithms. They did a segmentation of skin color that automatically represented colors, and the YCbCr color space is used because it frequently appears in the video template code and generates good human skin tincture effects. In addition, they consider CbCr plane for skin color distribution. People of different ethnicities have different skin tinctures and patterned designs. Deep learning techniques and multiple algorithms of Machine Learning are used to show translators translating between parties.

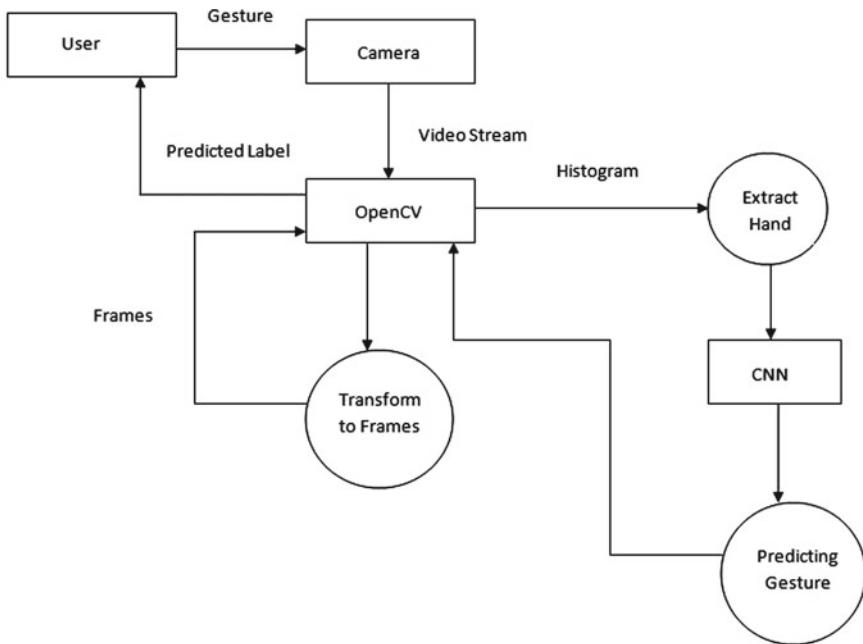
Nair et al. [3] selected a method for converting Malayalam into Indian Sign Language using virtual animation. The intermediate identification is used in this sign language approach by HamNoSys. The application in this method takes a set of words and forms them into animated parts. There is an interactive system that further transforms word parts into the structured design by HamNoSys. The Kerala government uses it to teach sign language and subtle awareness because it has become increasingly difficult to communicate with the general public while being deaf. As a result, its application strikes a balance between everything that has been built.

Mahesh et al. [4] have developed an application for the communication of deaf or hearing-aid people. Creating an application for such a community requires a lot of effort in terms of memory and fine design for implementation. The application is designed in a way to take a picture of a symbolic gesture and convert it into a meaningful word. At first, the gestures were compared using a histogram to test the sample which are then forced to BRIEF in such a way that there is a minimization in the weighting time on the CPU. Nowadays, the communication between normal people and the hearing-impaired has become a difficult task and the implementation of that makes society lack the right interpreter and an application for it.

Kumar et al. [5] selected a method for the Spanish-spoken language that translates the words to Spanish, which is great for deaf Spanish as it gives the ability to realize the language notation at such a fast rate that it will be converted to Spanish instead of commonly used English like ASL. The device created for the purpose includes multiple terms, such as a display interface used by hearing-impaired people to specify a string of symbol data, a translator, which can converts these strings to Spanish in a serialized formatted language, and a speech converter, which essentially converts that whole bit into a relevant statement in Spanish. The proposed work is focused on visual interface for the disabled people using sign language in real time.

### 3 Proposed Works

The following diagram shown in Fig. 1 explains the steps in execution to accomplish the purpose of the proposed work. It begins with an image acquisition which is used for image capturing in real time using OpenCV. It is followed by the extraction of various features of the image, mainly the boundary of the fingers. The proposed system employs a Convolutional Neural Network (CNN) model which comprises



**Fig. 1** Architectural diagram for sign language to sentence interpreter

two layers to determine the user's finger symbol. After the gestures are recognized, it is converted to text followed by sentence formation.

## 4 Implementation

The implementation of the proposed system consists of three phases, viz., dataset generation and acquisition, gesture recognition and classification, and sentence formation for the gestures which are discussed in the following Sects. 4.1, 4.2, and 4.3, respectively.

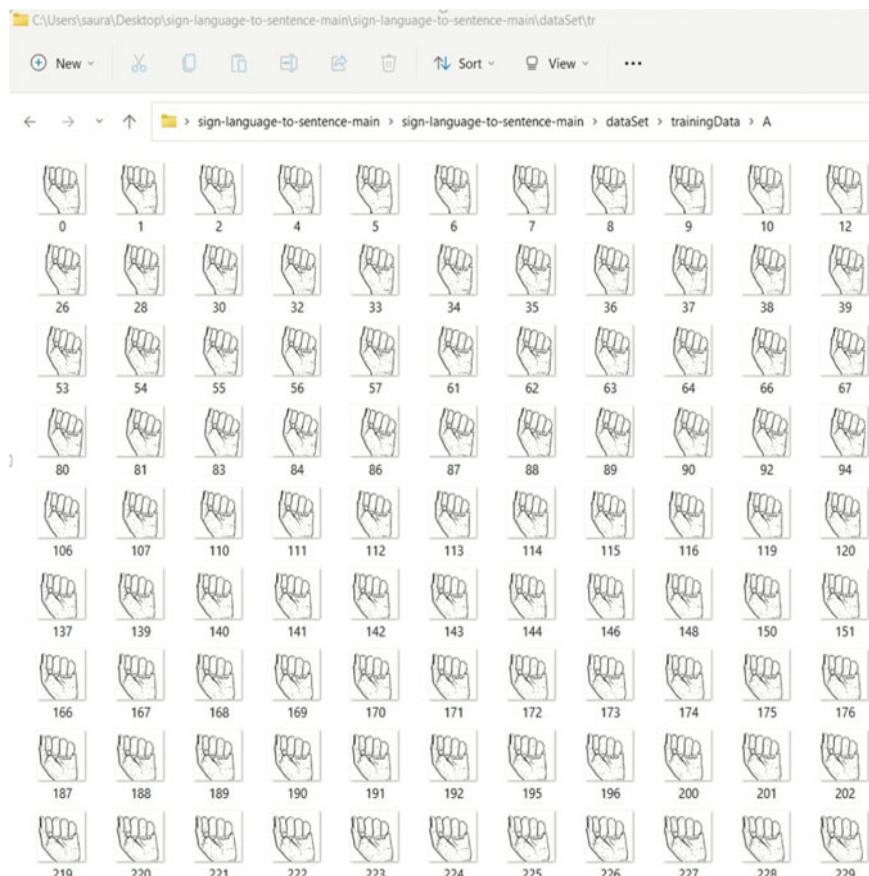
### 4.1 Dataset Generation and Acquisition

For the sign language to sentence interpreter, the aim is to find already made datasets, but there is no dataset that exists in the literature in the terms of unprocessed images that matches user requirements. The existing datasets were presented as RGB values. Hence, the custom dataset was created according to the requirement and the way of interpretation of the alphabet is based on ASL. The work has been carried out using

real-time data generation as per ASL. Figure 2 shows the sample dataset for the alphabet 'A'.

The various steps followed to create the dataset are as follows:

- (1) Open computer vision (OpenCV) library is used to generate the dataset. The dataset comprises totally 4160 pictures with respect to the 26 alphabets, 160 pictures for each. In this, 80% of the data has been considered as a training dataset (around 3328 pictures) and 20% has been considered for testing (around 832 pictures). Figure 3 shows the interface used to capture various datasets.
- (2) Next, the captured gesture is converted to grayscale. Figure 4 shows the resultant grayscale image of the captured gesture which was shown in Fig. 3.

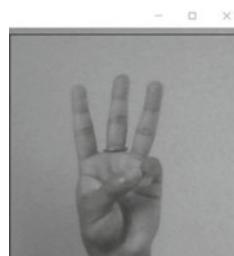


**Fig. 2** Sample dataset for character 'A'

**Fig. 3** Interface to capture raw image



**Fig. 4** Converting RGB image to grayscale



**Fig. 5** Image post-Gaussian Blur filter



- (3) Then, applying the Gaussian Blur filter to the grayscale image enables to extract the image's numerous features, mainly the boundary of fingers. The image, after applying Gaussian Blur, looks as follows in Fig. 5.

## 4.2 Gesture Recognition and Classification

Figure 6 describes the various steps that are followed in gesture classification. The gesture classification stage employs two algorithmic layers to forecast the user's final symbol. These two layers were used with increased accuracy.

### 4.2.1 Training and Testing

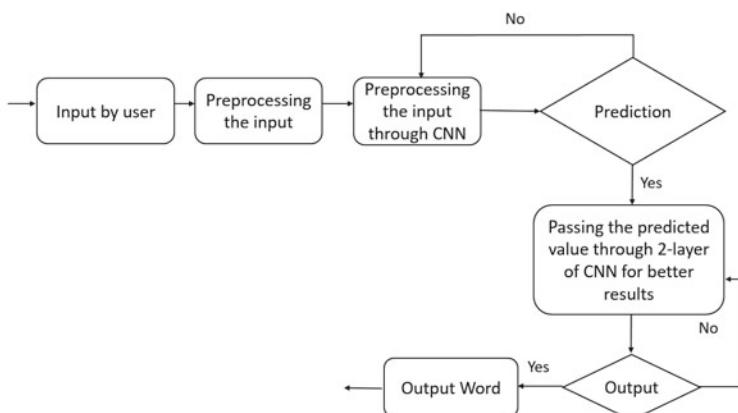
Gaussian Blur is used to remove unwanted noise after input images are transformed into grayscale images. An applied adaptive threshold technique is used to extract the hand out of the background and to make the images  $128 \times 128$  in size.

After applying all the techniques mentioned above, the model is ready to give the input images for training and testing. The prediction layer determines which classes the images will belong to. As a result, the output is scaled in the range of 0 and 1, such that the sum of every value in each class equals 1. This is done using the SoftMax activation function. The equation for the SoftMax function is calculated as shown below in Eq. (1).

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^p e^{x_j}}, \quad (1)$$

where  $\bar{x}$  is the input vector,  $e^{x_i}$  is the average exponential function for the input vector,  $p$  is the total number of classes,  $e^{x_j}$  is the average exponential function for the output vector. Here, the number of classes is 26 with respect to the alphabet; hence,  $p = 26$  and  $\bar{x}$  is the vector value of different signs for the specific alphabet.

Initially, the prediction layer's output will be quite different from the actual value. The network has been trained with labeled data to improve it. The performance



**Fig. 6** Flow diagram for gesture classification

evaluation method that is used in the classification is called cross-entropy. It is a linear function that is positive at values that differ from the labeled value and exactly zero at values that coincide with the labeled value. In order to optimize cross-entropy, the value was reduced as much as possible to close to zero value. Hence, the weights were adjusted in the neural networks to accomplish this at the network layer. A built-in function, namely TensorFlow, can be used to determine the cross-entropy. The cross-entropy was further optimized using a Gradient Descent optimizer, which is nothing but Adam Optimizer.

The layer description, corresponding activation function, and an optimizer utilized for the proposed sign language to sentence interpreter using Convolutional Neural Network (CNN) are as follows.

#### 4.2.2 Layers of CNN Model

The following are the layers of CNN model:

- (1) **First Convolution Layer:** The original (input) image is processed in the first convolution layer using 32 filter weights ( $3 \times 3$  resolution each), whereas the input image has a resolution of  $128 \times 128$  pixels. A  $126 \times 126$ -pixel image will be produced as a result, one for each filter weights.
- (2) **First Pooling Layer:** By using max pooling of  $2 \times 2$ , the images are down-sampled, which is the largest value kept in the array of size  $2 \times 2$  square. Hence, the image size is reduced to 63 by 63 pixels.
- (3) **Second Convolution Layer:** The  $63 \times 63$  image from the result of the first pooling layer is an input to the second convolutional layer. The 32 filter weights are applied during processing of the second convolution layer ( $3 \times 3$  pixels per weight). As a result, a  $60 \times 60$  resolution image will be formed.
- (4) **Second Pooling Layer:** The generated images are again down-sampled to  $30 \times 30$  resolution images using the max pool of  $2 \times 2$ .
- (5) **First Densely Connected Layer:** The images are given as an input to 128 neurons of fully connected layer. The convolution layer-2 output is reformed into an array of  $30 \times 30 \times 32$ , or 28,800 values. A 28,800-value array serves as the layer's input. These layers provide input to the second densely connected layer. To prevent over-fitting, a 0.5 dropout layer is used.
- (6) **Second Densely Connected Layer:** A fully connected layer with 96 neurons is fed information from the first densely connected layer.
- (7) **Final layer:** The input for this layer is taken from second densely connected layer. The amount of neurons used in this layer will be equivalent to the amount of classes (alphabets + blank symbol) that are classified.

#### 4.2.3 Activation Method

The activation method at the layer of fully connected neurons and convolutional was utilized by Rectified Linear Unit (ReLU). ReLU determines  $\text{Max}(x, 0)$  for each input pixel. The formula now has nonlinearity, which aids in understanding more complex features. Reduced computation time speeds up training and helps to eliminate the vanishing gradient issue.

With the help of the ReLU activation method, max pool size (2, 2) is applied to the input image. By reducing the number of parameters, both over-fitting and the cost of computation are reduced. The over-fitting issues have been solved with the usage of versatile data during the training and validation process.

#### 4.2.4 Optimizer

In function loss output, the model is updated using the Adam Optimizer. The adaptive gradient algorithm (ADAGRAD) and root mean square propagation expansions of two Gradient Descent methods are grouped in the Adam Optimizer.

The following algorithm outlines the functioning of the layer- 1.

##### Algorithm for CNN Layer-1

- (1) After feature extraction, to obtain the final result, the Gaussian Blur filter and threshold were used for the frame identified using OpenCV.
- (2) The finalized image is given to CNN for identification, and if a character is found throughout the course and if it is greater than 50 frames, then it is displayed and kept in the count when creating the word.
- (3) The space between words can be given only when blank symbol is identified.

#### 4.2.5 Layer- 2 of CNN Model

Two layers of algorithms were applied, to forecast and verify symbols that were extremely similar. The following symbols happen to be not displaying correctly, after the test that carried out.

- (1) U and R: D.
- (2) R and D: U.
- (3) K, D, I and T: I.
- (4) N and M: S.

As a result, three distinct classifiers for classifying these sets were developed to deal with the above cases:

- (1) {D, R, U}.
- (2) {T, K, D, I}.

**Fig. 7** Sentence formation in progress



(3) {S, M, N}.

The following algorithm outlines the functioning of the layer- 2.

#### Algorithm for CNN Layer-2

- (1) Find a variety of sets of symbols that, upon detection, produce comparable outcomes.
- (2) Next, classify the sets using classifiers.

### 4.3 Sentence Formation for the Gestures

This section describes about the word and sentence formation phase. At any time, the number of letters identified goes beyond a particular value and any other letter is not close to it by a threshold, then output the letter and attach it to the current string (value is 50 and difference threshold is 20). Otherwise, to reduce the likelihood of predicting the incorrect letter, the present symbol (letter) loaded in the dictionary will be erased. If the current buffer is empty and the number of blank (plain background) detections exceeds a certain value, no spaces are found. In other instances, it prints a space to represent when a word will end, and the current word is added to the sentence below, as shown in Fig. 7.

#### 4.3.1 Auto Correction Feature

A Python package called Hunspell provides appropriate substitutes for each input word and outputs a list of related words to the current word, letting the user can choose a word to add to the current sentence. This supports in minimization of errors in the spellings and also helps in resolving complex words.

## 5 Result and Discussion

Figures 8 and 9 demonstrate the confusion matrices for the proposed system. Confusion matrices give a complete view of how well our classification model is performing.

From the confusion matrix, the accuracy (in %) is calculated as shown below in Eq. (2).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y			
A	147	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	2	0	0			
B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0			
C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
D	0	0	0	145	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0			
E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
F	0	0	0	0	0	135	0	0	0	0	0	0	4	0	0	0	0	0	0	1	0	0	2	10	0			
G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0			
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0			
r	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	7	1	0	0	0	0			
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0			
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0			
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0			
N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0			
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0	0		
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0		
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0		
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0		
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	8	0	0		
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	151	0	0	0	0	0	0		
U	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	0	115	0	0	0	0	0	0		
V	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0	0	0	0	
W	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0	0	0	0	0
X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	148	0	
Y	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Fig. 8** Confusion matrix of using Algorithm 1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y						
A	147	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	2	0	0					
B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0					
C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
D	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
F	0	0	0	0	0	135	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	10	0	0	0					
G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0					
o	H	1	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1					
r	I	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0					
r	J	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
e	K	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
c	L	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0					
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0					
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0	0	0				
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0					
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0					
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0					
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	10	0	0	0	0	133	0	0	0	8	0	0	0				
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0	0	0			
U	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0		
V	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0	0	0	0	0	
W	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0	0	0	0	0	0	
X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Y	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Z	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Fig. 9** Confusion matrix of using Algorithm 1 and Algorithm 2

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total Number of Predictions}} * 100. \quad (2)$$

In Figs. 8 and 9, the diagonal elements represent the various correct predictions and off-diagonal elements represent the misclassifications. Hence, based on Eq. (1), the classification model reached 95.8% of accuracy using algorithm of layer-1 only and achieved 98.0% of accuracy when layer-1 and layer-2 are combined together. Similarly, the precision, recall, and *F1*-score are calculated as shown below in Eq. (3).

$$\text{Precision} = \frac{Tp}{Tp + Fp}, \quad (3)$$

where *Tp* and *Fp* are true positives and false positives, respectively. The precision value obtained was 92% in this work.

The recall or true-positive rate (TPR) for the proposed work is calculated as shown below in Eq. (4).

$$\text{Recall} = \frac{Tp}{Tp + Fn}, \quad (4)$$

where *Fn* is a false negative. The recall of 95.8% was achieved by using the layer-2 of CNN algorithm.

The *F1*-score is computed based on the obtained confusion matrix which is calculated as shown below in Eq. (5).

$$\text{F1-score} = \frac{2Tp}{2Tp + Fp + Fn}. \quad (5)$$

The proposed algorithm obtains the *F1*-score of 94%. The state-of-the-art deep learning algorithms versus proposed CNN2 layer model is shown below in Table 1.

The proposed solution was tested in real-world scenarios, where it was demonstrated that the obtained classification models were able to recognize all of the trained gestures. After implementing two layers of algorithms, the prediction was also improved. This provides the ability to determine nearly all the symbols provided which are shown properly.

**Table 1** State-of-the-art deep learning algorithms versus CNN model

Algorithms	Accuracy (%)	Precision	Recall	F1-score
RNN	92.13	0.84	0.83	0.81
Yolo	93.04	0.87	0.89	0.88
CNN1-layer	95.8	0.90	0.88	0.89
CNN2-layer	98	0.92	0.95	0.94

## 6 Conclusion and Future Enhancements

Sign language to sentence interpreter provides a simple demonstration to solve problems in computer vision with an immensely high level of precision with the use of CNN. The sign language translator for finger spelling is acquired 98% accurate result. A vision-based system that can translate American Sign Language hand gestures into text or speech was presented in this paper. By creating the necessary dataset and training the CNN, the proposed system can be extended to include additional sign languages. The gesture recognition system could be expanded to include all ASL alphabets as well as non-alphabet gestures as possible extensions of this project. It also might be developed as a web or mobile application in the future to make it easier for users.

## References

1. Kau L, Su W, Yu P, Wei S (2015) A real-time portable sign language translation system. In: 2015 IEEE 58th international midwest symposium on circuits and systems (MWSCAS), FortCollins, CO, 2015, pp 1-4. <https://doi.org/10.1109/MWSCAS.2015.7282137>
2. Shahriar S et al (2018) Real-time american sign language recognition using skin segmentation and image category classification with convolutional neural network and deep learning. In: TENCON 2018–2018 IEEE Region 10 conference, Jeju, Korea (South), pp 1168–1171. <https://doi.org/10.1109/TENCON.2018.8650524>
3. Nair MS, Nimitha AP, Idicula SM (2016) Conversion of Malayalam text to Indian sign language using synthetic animation. In: 2016 international conference on next generation intelligent systems (ICNGIS), Kottayam, 2016, pp 1–4. <https://doi.org/10.1109/ICNGIS.2016.7854002>
4. Mahesh M, Jayaprakash A, Geetha M (2017) Sign language translator for mobile platforms. In: 2017 international conference on advances in computing, communications and informatics (ICACCI), Udupi, pp 1176–1181. <https://doi.org/10.1109/ICACCI.2017.8126001>
5. Kumar SS, Wangyal T, Saboo V, Srinath R (2018) Time series neural networks for real time sign language translation. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA), Orlando, FL, pp 243–248. <https://doi.org/10.1109/ICMLA.2018.00043>.
6. Kelly D, Mc Donald J, Markham C (2011) Weakly supervised training of a sign language recognition system using multiple instance learning density matrices. IEEE Trans Syst Man Cybernet Part B (Cybernetics) 41(2):526–541. <https://doi.org/10.1109/TSMCB.2010.2065802>
7. Jimenez J, Martin A, Uc V, Espinosa A (2017) Mexican sign language alphanumerical gestures recognition using 3D Haar-like features. IEEE Lat Am Trans 15(10):2000–2005. <https://doi.org/10.1109/TLA.2017.8071247>
8. Mohandes M, Deriche M, Liu J (2014) Image-based and sensor-based approaches to arabic sign language recognition. IEEE Trans Human-Mach Syst 44(4):551–557. <https://doi.org/10.1109/THMS.2014.2318280>
9. San Segundo R, Gallo B, Lucas JM, Barra-Chicote R, D’Haro LF, Fernandez F (2009) Speech into sign language statistical translation system for deaf people. IEEE Latin America Trans 7(3):400–404. <https://doi.org/10.1109/TLA.2009.5336641>
10. Lopez-Ludena V, San-Segundo R, Martin R, Sanchez D, Garcia A (2011) Evaluating a speech communication system for deaf people. IEEE Lat Am Trans 9(4):565–570. <https://doi.org/10.1109/TLA.2011.5993744>

11. Krak I, Kryvonos I, Wojcik W (2012) Interactive systems for sign language learning. In: 2012 6th International conference on application of information and communication technologies (AICT), Tbilisi, pp 1–3. <https://doi.org/10.1109/ICAICT.2012.6398523>
12. Abraham E, Nayak A, Iqbal A (2019) Real-time translation of indian sign language using LSTM. In: 2019 Global conference for advancement in technology (GCAT), Bangaluru, India, pp 1–5. <https://doi.org/10.1109/GCAT47503.2019.8978343>
13. Hegde B, Dayananda P, Hegde M, Chetan C (2019) Deep learning technique for detecting NSCLC. *Int J Recent Technol Eng (IJRTE)* 8(3):784
14. Kasinathan V et al. (2021) Sign language translation system using convolutional neural networks approach. In: Advances in mechatronics, manufacturing, and mechanical engineering. Springer, Singapore, pp 161–171
15. Mukherjee H et al. (2022) Automatic sign language identification using convolutional neural network. In: Computational intelligence in pattern recognition. Springer, Singapore, pp 293–302
16. Bhagwat A, Gupta P, Kadam N (2022) Sign language to categorical text using convolutional neural network. In: 2022 10th International conference on emerging trends in engineering and technology-signal and information processing (ICETET-SIP-22), IEEE, pp 1–6
17. Kumar A, Madaan M, Kumar S, Saha A, Yadav S (2021) Indian sign language gesture recognition in real-time using convolutional neural networks. In: 2021 8th International conference on signal processing and integrated networks (SPIN), IEEE, pp 562–568
18. Chandrasekaran S (2021) American sign language recognition and translation using deep learning and computer vision. Dissertations Dublin, National College of Ireland, pp 1–18
19. Jayasekera MKC (2021) Real-time New Zealand sign language translator using convolution neural network. Dissertations The University of Waikato, pp 1–98
20. Rawf KH, Abdulrahman A, Mohammed A (2022) Effective Kurdish sign language detection and classification using convolutional neural networks. pp 1–19

# A Comparative Study of Deep Learning-Based Face Recognition and Emotion Detection Techniques Using Social Media Customized Cartoon Post



Jayanta Paul, Anuska Roy, Siddhartha Mallick, and Jaya Sil

**Abstract** Cartoon posts regarding many renowned Indian personalities have increased alarmingly in different social media platforms. Consequently, it throws challenges for recognizing and identifying the personalities and analyzing the features of their faces for understanding the general emotion conveyed by the post. In this work, we have created a manually labeled dataset using Indian cartoon posts and investigating different approaches using deep learning models to recognize the face of the person and understand the mood of the person from the post. In addition, we proposed a pipeline called “Haar base ViT fully connected face recognition and person identification along with emotion detection” using vision transformer. The performance of the proposed pipeline is compared with three pretrained CNN based models (MobileNetV2, VGG-16, and ResNet-50). Our approach exhibits competitive results (in terms of accuracy, it is 90% for face recognition and 88% for emotion detection) when evaluated on the individual tasks.

**Keywords** Social media analysis · Deep learning · Face recognition · Emotion detection · Vision transformer

---

J. Paul (✉) · A. Roy · S. Mallick · J. Sil

Indian Institute of Engineering Science and Technology, Shibpur 711103, India

e-mail: [2020csp003.jayanta@students.iests.ac.in](mailto:2020csp003.jayanta@students.iests.ac.in)

A. Roy

e-mail: [2020csb017.anuska@students.iests.ac.in](mailto:2020csb017.anuska@students.iests.ac.in)

S. Mallick

e-mail: [2021csm008.siddhartha@students.iests.ac.in](mailto:2021csm008.siddhartha@students.iests.ac.in)

J. Sil

e-mail: [js@cs.iests.ac.in](mailto:js@cs.iests.ac.in)

## 1 Introduction

With the rapid growth of social media [14] usages, users often share their opinions [23] using cartoon posts regarding different domains, like social, political, economical, cultural, and others. We observe that the cartoon faces are usually derived from respective actual face images for different purposes such as entertainment, criticism, praising etc. We have investigated pretrained CNN models MobileNetV2 [16], VGG-16 [18], ResNet-50 [7] and propose the pipeline based on vision transformer(ViT) model [4] to recognizing the faces, and the persons are identified from the cartoon posts. Detecting emotion is a special thrust area of research from the past few years, and facial emotions are meant to non-verbal communications. Emotion detection normally includes facial gesticulations [8, 11]. In the paper, we studied emotion detection tasks of the works [10, 21], and the proposed pipeline is employed to detect the emotions from the recognized faces. People in social media often depict cartoon faces with different types of emotions of the same character. The enthusiasm behind such contemporary research lies upon the availability of abundant animated cartoons, which provide an opportunity to the researchers to detect the emotions and understand whether it is positive or negative kind of information scattering.

The paper is organized into seven sections. Section 2 consists of our main contributions in the domain of face recognition and emotion detection. Section 3 describes the literature survey in this field, while Sect. 4 presents the proposed cartoon dataset. Section 5 describes the proposed pipeline of work, and in Sect. 6, the results of our proposed pipeline ( $HVFF_R E_D$ ) are compared with other three CNN models. Section 7 concludes the paper.

## 2 Contribution

Our work focuses on

- Preparation of cartoon dataset and labeled manually for both the tasks of face recognition, vis-a-vis person identification and emotion detection.
- Cartoon posts are analyzed to recognize the faces and identify the corresponding persons along with extraction of facial expressions using the proposed pipeline ( $HVFF_R E_D$ ).
- A comparative study is provided among  $HVFF_R E_D$  and other CNN-based models [17].

## 3 Related Work

Over the past few years, the exponential growth of social media usage leads to increasing the need of understanding the content of the social media posts. Sentiment and emotion expressed in social media posts while analyzed give information regarding

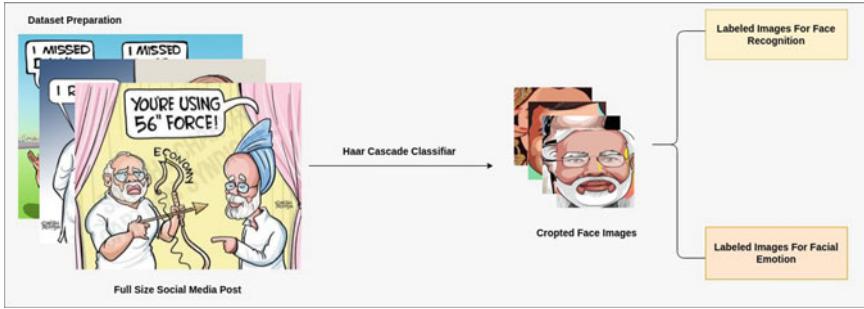
outlook of the users, in connection with any event, person concerned, etc. In this context, deep learning-based face recognition and emotion detection methods have emerged as an important area of research [12]. There are many related works [3, 15] in the field of face recognition and emotion detection [5, 22] using CNN models [2, 6]. Liu et al. [9] proposed a deep learning approach for emotion detection from facial expression recognition. The authors used a customized cartoon post as the training data to train a deep neural network (DNN) model. The results showed that the DNN model outperformed other traditional machine learning algorithms in terms of accuracy and efficiency. Rajat et al. [1] proposed a multimodal deep framework for derogatory social media post identification of a recognized person. Jayanta et al. [13] proposed IoT system framework of low-computation for the purpose of face recognition using deep learning algorithm. In the paper, we have used transformer model (vision transformer/ViT) [4] to propose the pipeline for face recognition and emotion detection. In particular, customized cartoon post has been used as a unique feature to improve performance of the face detection and emotion detection tasks.

## 4 Dataset Preparation

The proposed Social-Cartoon-Dataset (named as: Soc-toon-set) is built considering the cartoon posts from popular social media Websites like Facebook, Twitter, Instagram, etc. Haar cascade classifier [20] is used to identify the cropped faces of the cartoon images. Figure 1 represents the dataset preparation pipeline. The images are labeled in two ways for two different tasks (face recognition and emotion detection). First, for face recognition, the images are labeled manually into 24 classes according to the names of socially famous persons (e.g., Sachin Tendulkar, Lata Mangeshkar, Sundar Pichai, etc.) present in the post. For emotion detection of the posts, we manually labeled the images into 4 classes, i.e., angry, happy, neutral, and sad. There is class imbalance in the Soc-toon-set, therefore in order to increase the number of instances of those classes with fewer images, we have performed data augmentation (mainly rotation, change in brightness, zoom, height, and width shift).

## 5 Methodology

The proposed methodology to recognizing the faces of famous personalities and predicting their emotions using deep learning techniques is presented in this section. Our main focus is using the state-of-the-art transformer-based model called vision transformer [4] for the tasks. We have used pretrained CNN models for comparing performance with the vision transformer model using the proposed cartoon dataset.



**Fig. 1** Dataset preparation pipeline

### 5.1 MobileNetV2

MobileNetV2 [16] is a convolutional neural network (CNN) consisting of 53 layers and capable of learning feature representations efficiently for a variety of images. The model takes an image as input of size of  $224 \times 224$ . The model has CNN-based blocks containing skip connections. Each block first expands feature channels of the input to a high-dimensional representation and filters it with a light-weight depth-wise convolution. Next, the feature channels are reverted back to a lower dimension for maintaining an inverted bottleneck design. The residual connections are attached to the bottleneck.

### 5.2 VGG-16

VGG-16 [18] is a 16 layer deep CNN model whose depth is increased by the use of an architecture containing tiny kernels of size  $3 \times 3$ . There are two variants of VGGNet, one containing 16 layers and the other containing 19 layers. For our experiments, we use the first variant dubbed VGG-16 having approximately 135 million trainable parameters. The model has proven a successful deep CNN architecture which gives impressive results on various image classification problems.

### 5.3 Resnet-50

ResNet-50 [7] is a CNN model containing 50 layers. The model constituting an initial stem operation achieved by convolutions of size  $7 \times 7$  with a stride of 2 followed by a series of convolutions with skip connections in between for enhancing feature response quality. ResNet-50 contains  $3 \times 3$  convolutions with skip connections having a stage compute ratio of 3:4:6:3. The model has a floating point operations per

second (FLOPs) value of 3.8 billion which is less than that of VGG-16 (15.3 billion). We have used average pooling and a fully connected neural network in order to classify emotions and recognize faces.

## 5.4 Vision Transformer (ViT)

Vision transformer [4] is a state-of-the-art transformer model mainly employed for classification in computer vision. It comprises of two parts, such as linear projection of flattened patches (LPFP) and a standard transformer encoder. In essence, ViT is primarily adopted as a computer vision model, but the core working is that of a transformer encoder used in natural language processing. Thus, the LPFP module can be thought of as a tokenizer which produces processed input embeddings for the encoder model in order to extract learned embeddings for classification. The input data fed to the transformer model is provided in the form of two-dimensional images by the LPFP module. For this purpose, image considered as input is of height  $H$ , width  $W$ , and number of channels  $C$ . The image is split into small-scaled two-dimensional patches results into  $N = HW/P^2$  number of patches, where each patch has a resolution of  $(P, P)$  pixels.

## 5.5 Haar Base ViT Fully Connected Face Recognition and Person Identification Along with Emotion Detection(HVFF<sub>R</sub>E<sub>D</sub>)

The proposed pipeline, named “Haar base ViT fully connected face recognition and person identification along with emotion detection(HVFF<sub>R</sub>E<sub>D</sub>)”, contains a transformer-based pretrained model called the vision transformer (ViT). One theoretical advantage of ViT over the three pretrained CNN-based models mentioned above is that ViT uses transformer layers which contain multi-headed self attention. This gives an upper hand to ViT when it comes to representing semantic features accurately since attention mechanisms are designed to suppress unwanted feature noise in representations and to highlight important semantics from the map. However, due to a lack of attention mechanisms in pretrained CNN-based models, they cannot achieve the intricacies and nuances that self-attention in ViT provides. Our proposed HVFF<sub>R</sub>E<sub>D</sub> pipeline works according to the algorithmic steps given below:

**Step 1:** Each image of size  $400 \times 320$  pixels is passed through a pretrained Haar cascade classifier. The classifier detects faces (if any) and crops them.

**Step 2:** The cropped images, i.e., only the faces are resized to a dimension of  $182 \times 182$  pixels.

**Step 3:** The face images having  $N = HW/P^2$  number of patches.

**Step 4:** Every patch is flattened into vector  $x_r^n$  of size  $P^2 \times C$  where  $n = 1, \dots, N$ .

**Step 5:** The flattened patches are mapped to patch embeddings of  $D$  dimension using trainable linear projection  $E$ , results an ordered set of embedded image patches.

**Step 6:** Our pipeline prepend a learnable class embedding,  $x_{\text{class}}$ , to the sequence of embedded image patches. The value of  $x_{\text{class}}$  in the last layer of the encoder is fed to a MLP head for facial recognition and emotion detection.

**Step 7:** The patch embeddings are finally augmented with one-dimensional positional embeddings,  $E_{\text{pos}}$ , introducing positional information into the input, which is also learned during training. The sequence of embedding vectors  $\text{EV}^0$  that results from the aforementioned operations is the following:

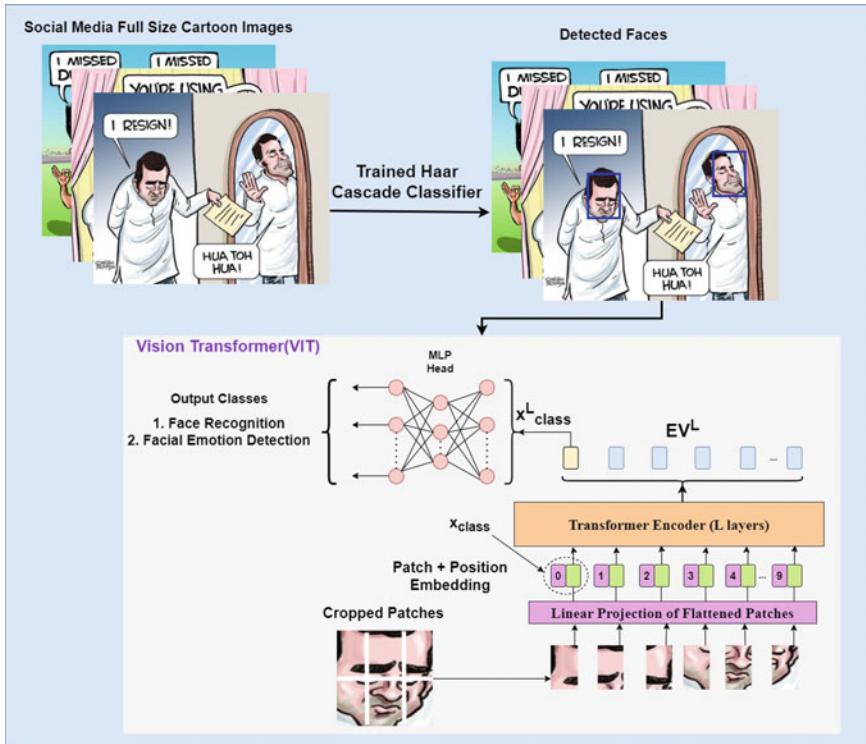
$$\text{EV}^0 = [x_{\text{class}}, x_{\text{PE}}^1, \dots, x_{\text{PE}}^N] + E_{\text{pos}}$$

**Step 8:** In order to perform classification, we feed  $\text{EV}^0$  at the input of the transformer encoder consisting of  $L$  identical layers having architecture similar to the encoder of the original transformer model [19]. The  $L^{\text{th}}$  layer of the encoder produces a sequence of learned embedding vectors  $\text{EV}^L$ . The first element of  $\text{EV}^L$  contains the learned embedding patch of  $x_{\text{class}}$  which now can be represented as  $x_{\text{class}}^L$  in the  $L^{\text{th}}$  layer.  $x_{\text{class}}^L$  is fed to a multi-layer perceptron (MLP) layer with a zero initialized softmax unit for obtaining class predictions for both the tasks of face recognition and emotion detection, i.e., one pipeline ( $\text{HVFF}_R E_D$ ).

The above algorithm has proven to be successful in providing more accurate results for face recognition and emotion detection using cartoon faces when compared with CNN-based models. Figure 2 shows the proposed pipeline of  $\text{HVFF}_R E_D$ .

## 6 Experimental Analysis

Recognition of faces and classifying emotions from the cartoon faces is a multi-class classification problem. The two classification tasks are performed simultaneously using four models, namely MobileNetV2, ResNet-50, VGG-16 and our proposed pipeline  $\text{HVFF}_R E_D$ . Tables 1 and 2 show that for both the tasks,  $\text{HVFF}_R E_D$  pipeline has outperformed the other three CNN models. Considerably, high accuracy of 90% for face recognition along with person identification and 88% for emotion identification are achieved. The metrics precision, recall and F1-score of our proposed pipeline ( $\text{HVFF}_R E_D$ ) have surpassed all the CNN-based models. Next to our proposed pipeline ( $\text{HVFF}_R E_D$ ), ResNet-50 has performed decently in terms of accuracy 77% for face recognition and 81% for emotion detection. Other two models MobileNetV2 and VGG-16 have also given satisfactory results, but they are not at par with the other two models.



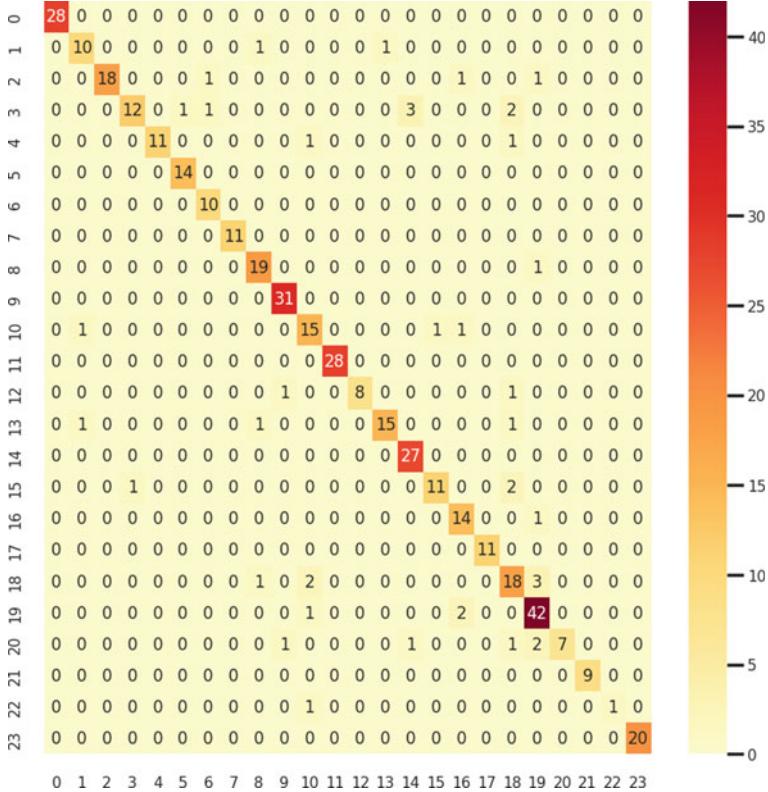
**Fig. 2** Haar base ViT fully connected face recognition and person identification along with emotion detection (HVFF<sub>R</sub> E<sub>D</sub>)

**Table 1** Experimental analysis for face recognition

Model	Accuracy	Precision	Recall	F1	AUC
HVFF <sub>R</sub> E <sub>D</sub>	<b>0.9005</b>	<b>0.9066</b>	<b>0.9005</b>	<b>0.8984</b>	<b>0.9950</b>
MobileNetV2	0.7200	0.5800	0.6000	0.5900	0.5765
ResNet-50	0.7755	0.7715	0.7755	0.7676	0.7862
VGG-16	0.6800	0.5677	0.5845	0.5700	0.5562

**Table 2** Experimental analysis for facial emotion detection

Model	Accuracy	Precision	Recall	F1	AUC
HVFF <sub>R</sub> E <sub>D</sub>	<b>0.8807</b>	<b>0.8769</b>	<b>0.8807</b>	<b>0.8783</b>	<b>0.9800</b>
MobileNetV2	0.7809	0.6765	0.6347	0.5981	0.5866
ResNet-50	0.8165	0.8182	0.8165	0.8173	0.8235
VGG-16	0.7623	0.7000	0.7200	0.7000	0.5656

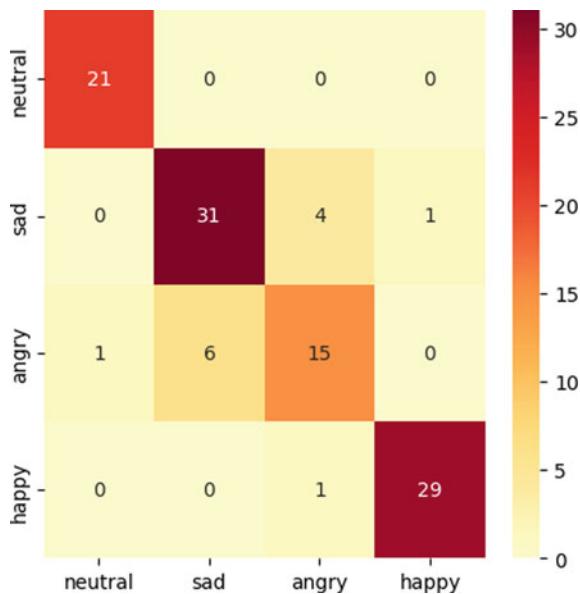


**Fig. 3** HVFF<sub>R</sub> E<sub>D</sub> confusion matrix for face recognition

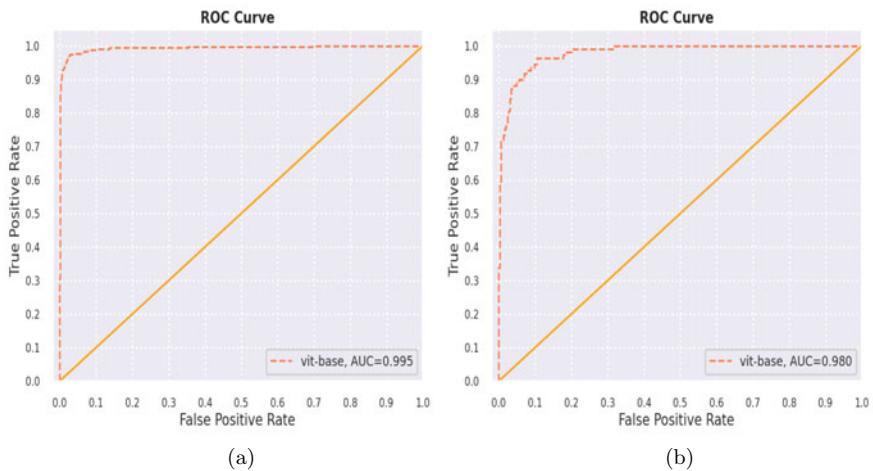
The confusion matrices shown in Figs. 3 and 4 represent how the model HVFF<sub>R</sub> E<sub>D</sub> accurately predicts maximum test cases for different classes. This type of prediction exhibits that our proposed model can easily identify the cartoon faces from the posts in social media and expresses the emotion of the recognized face (i.e., angry, happy, sad, and neutral).

The high AUC score for face recognition: 0.9950 and emotion detection: 0.9800 tells that the vision transformer model is capable enough to distinguish between the class labels of both facial recognition and emotion detection. This proposed pipeline has an AUC near to one implies, and it has a well measurement of the separability.

Figure 5a, b shows HVFF<sub>R</sub> E<sub>D</sub> ROC curve for (a) face recognition and (b) facial emotion detection.



**Fig. 4** HVFF<sub>E\_D</sub> confusion matrix for facial emotion detection



**Fig. 5** HVFF<sub>E\_D</sub> ROC curve for **a** face recognition and **b** facial emotion detection

## 7 Conclusion

In the paper, we have proposed the pipeline of Haar base ViT fully connected face recognition along with person identification and emotion detection( $HVFF_R E_D$ ) tasks. Considering the cartoon posts of famous Indian personalities, we have also prepared a new dataset (Soc-toon-set) and labeled each instance manually by their names and the emotions each face convey. Our proposed pipeline shows better performance when compared to other three CNN-based models. The proposed pipeline can be utilized in real-time applications provided that the computer system has adequate GPU support for carrying out a large number of floating point operations in a matter of seconds. It can further be used in automation of driver monitoring system (GPU supported) or in security purpose to identify any person (by training the pipeline with some other dataset). However, our method requires significant computer resources to operate in real-time, which can be considered as a drawback of our approach. Nevertheless, we believe that our work will give rise to a plethora of techniques to find out gestures from facial emotions and what kind of information a social media post is spreading through cartoon memes.

## References

1. Bhowmick RS, Ganguli I, Paul J, Sil J (2021) A multimodal deep framework for derogatory social media post identification of a recognized person. *Trans Asian and Low-Res Lang Inf Proc* 21(1):1–19
2. Dachapally PR (2017) Facial emotion detection using convolutional neural networks and representational autoencoder units. *ArXiv preprint arXiv:1706.01509*
3. Daros AR, Zakzanis KK, Rector NA (2014) A quantitative analysis of facial emotion recognition in obsessive-compulsive disorder. *Psychiatry Res* 215(3):514–521
4. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S et al (2020) An image is worth  $16 \times 16$  words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*
5. Jain N, Gupta V, Shubham S, Madan A, Chaudhary A, Santosh K (2021) Understanding cartoon emotion using integrated deep neural network on large dataset. *Neural Comput Appl* 1–21
6. Khan AR (2022) Facial emotion recognition using conventional machine learning and deep learning methods: current achievements, analysis and remaining challenges. *Information* 13(6):268
7. Koonce B, Koonce B (2021) Resnet 50. Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization, pp 63–72
8. Küntzler T, Höfling TTA, Alpers GW (2021) Automatic facial expression recognition in standardized and non-standardized emotional expressions. *Front Psychol* 12:1086
9. Li M, Xu H, Huang X, Song Z, Liu X, Li X (2018) Facial expression recognition with identity and emotion joint learning. *IEEE Trans Affect Comput* 12(2):544–550
10. Mellouk W, Handouzi W (2020) Facial emotion recognition using deep learning: review and insights. *Proced Comput Sci* 175:689–694
11. Minaee S, Minaei M, Abdolrashidi A (2021) Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors* 21(9):3046
12. Paul J, Bhowmick RS, Sen R, Ray D, Manjhi SS, Sen S, Sikdar BK (2020) Evaluation of face recognition schemes for low-computation iot system design. In: 2020 24th international symposium on VLSI design and test (VDAT). IEEE, pp 1–6

13. Paul J, Bhowmick RS, Sil J (2022) Low-computation iot system framework for face recognition using deep learning algorithm. In: Computational intelligence in pattern recognition: proceedings of CIPR 2022. Springer, pp 24–35
14. Perrin A (2015) Social media usage. Pew Res Center 125:52–68
15. Roopa N (2019) Emotion recognition from facial expression using deep learning. Int J Eng Adv Technol (IJEAT) ISSN 2249–8958
16. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
17. Saroop A, Ghugare P, Mathamsetty S, Vasani V (2021) Facial emotion recognition: a multi-task approach using deep learning. ArXiv preprint [arXiv:2110.15028](https://arxiv.org/abs/2110.15028)
18. Tammina S (2019) Transfer learning using vgg-16 with deep convolutional neural network for classifying images. Int J Sci Res Publ (IJSRP) 9(10):143–150
19. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. <https://doi.org/10.48550/ARXIV.1706.03762>. <https://arxiv.org/abs/1706.03762>
20. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vis 57(2):137–154
21. Zhang H, Jolfaei A, Alazab M (2019) A face emotion recognition method using convolutional neural network and image edge computing. IEEE Access 7:159,081–159,089 (2019)
22. Zhang S, Liu X, Yang X, Shu Y, Liu N, Zhang D, Liu YJ (2021) The influence of key facial features on recognition of emotion in cartoon faces. Front Psychol 12(687):974
23. Zhao Z, Zhu H, Xue Z, Liu Z, Tian J, Chua MCH, Liu M (2019) An image-text consistency driven multimodal sentiment analysis approach for social media. Inf Proc Manage 56(6):102,097

# A Hybrid GA-PSO based approach for Mining Top-Ranked Web Pages to Reorganize Websites



Santosh Kumar, Tejas Kesarwani, and Sumit Kumar

**Abstract** There are millions of websites in the world, and these are increasing gradually. Locating pertinent information from the Internet is a very challenging job. Most of the websites consist of several web pages. These web pages may reach from hundreds to thousands in a single website. It is a difficult task for website designer as well as owners of the website to organize these large number of web pages so that website becomes interactive, and navigation in the website becomes easier. At the same time, required information could be easily obtained to the users. It is possible only if we could bank bees web pages based on the parameters like access frequency, unique visitors to a web page, active time spent by the user, common keywords given by the user, hubs, and authority count values. In this paper, the problem of ranking these web pages has been addressed by the hybrid GA-PSO algorithm where the web pages are first ranked by GA algorithm, and thereafter, these ranked optimal web pages are further refined by the PSO algorithm. Experimental evidence has shown that the proposed hybrid approach allows to find out the top-ranked ‘ $K$ ’ documents to reorganize the structure of website.

**Keywords** Genetic algorithm · PSO · Metaheuristic · Website reorganization

## 1 Introduction

Web browsers turned into an instrument to make the data accessible at our fingertips. As years passed, size of Internet became huge in data, and it turned out to be difficult to get information as per the need. Web mining came as a salvage for the above issue. Web mining alludes to the general process of finding possibly helpful and already obscure data or information from the web information. Web mining is used to collect crucial data, create new information from relevant data, personalize the

---

S. Kumar (✉) · T. Kesarwani · S. Kumar

Department of Computer Science and Engineering, Galgotias University, Greater Noida, Uttar Pradesh 201310, India

e-mail: [santoshg25@gmail.com](mailto:santoshg25@gmail.com)

data, and learn about customers or specific clients and a few other things. Web mining techniques have been abundantly used in the literature as well as in real applications to overcome the problem of information overload. In order to reduce the amount of time required for calculations to address the efficiency issue, an interrogative technique has also been presented. The study in [1, 2] proposed different techniques of nature inspired algorithm to redesign website architectures. The proposed calculation is tried widely with mathematical models. Transformative processing techniques help in creating web mining apparatuses which extricate significant and required data [3]. Moreover, an observational review with a certifiable site is led to check the calculation materialness.

The genetic algorithm (GA) is a method that relies on natural selection, the process that fuels evolution, to address both limited and unconstrained optimization problems. Over and over, the genetic algorithm modifies a population of unique configurations. The genetic algorithm selects individuals from the current population to serve as guardians at each stage and uses them to produce the offspring of the future. The population ‘evolves’ toward the ideal system as time goes on. The genetic algorithm may be used to solve a variety of optimization challenges that are inapplicable to traditional methods, such as those where the target ability is discontinuous, non-differentiable, unpredictable, or fundamentally nonlinear. When just a few sections of a blended number programs are allowed to be whole number esteemed, the genetic algorithm can address the problem. Both textual and graphical data are contained in the information used for online content mining [4–7]. According to the searches, there are two types of content mining. These are search result mining and website page content mining. The practice of content-based online search is known as website page mining. The pages from the previous search are further examined by search result content mining. Web content mining techniques have been used evolutionarily. The suggested method makes advantage of several genetic algorithm strategies for online content mining. GA is a search algorithm based on genetics and natural selection. The main definition of GA is survival of the fittest, often known as normal determination [1, 8]. It differs from the previous search algorithms because it searches across populations of focused objects and employs boundary set coding rather than actual boundary values. Regular item-set mining assumes a significant part in a few information storage of association rules, connections, grouping of highly multi layered natural data, and organization which are all mining topics. The number of item-sets that might be produced from a dataset, ‘ $d$ ’ with ‘ $k$ ’ items, omitting the empty set, is  $2^k - 1$ . The support of every information set should be recorded by looking at each and every occurrence in the dataset in order to look the regular item-sets. Because of the large number of frequent item-sets whose supports counts must be computed, a brute force method will be computationally expensive.

Particle swarm optimization (PSO) is an algorithm that draws its inspiration from groups of moving fish or flocks of birds. Particles and swarms are the terms used to describe the population in PSO. Based on particle mobility and intelligence, each and every particle searches for the best spot. Therefore, the goal of every particle motion is to determine its present location, its best location, and the sum of its best

locations. The fitness function, which is derived from the fitness value, estimates the particle's present position.

The general use of web search engines is decreasing as the size of the web is continuously growing. Many search engines are available to get the required and relevant information from the web. Many approaches based on randomized algorithms like iterative improvement (II) [2], simulated annealing (SA) [9], genetic algorithm [5, 8, 9], and memetic algorithm [1] have already been used to rank the web pages. These search engines used a few features to rank the pages and display these web pages in response to user's query. In this paper, many relevant features based on web usage, structure, and content data have been used to rank the web pages for reorganizing the structure of website. These features are keyword frequency, hubs and authority values, active time spend by the user on any web page, number of unique visitors, and access account. The ranking function uses all these features to prioritize web pages. This ranking function makes the proposed approach a novel contribution in the literature. Further, the hybrid approach GA followed by PSO is able to select the better set of top-'K' ranked web documents useful to modify the structure of any website.

## 2 Related Work

In [10], an approach for reorganizing websites using ant colony-based approach has been proposed. However, finding the ideal solution using the heuristic technique incorporating two subsequent 0–1 programming models still takes a very lengthy amount of time to compute, particularly when the site has several hyperlinks. Marghny et al. [11] have suggested a way for using genetic algorithms to mine online material. Genetic algorithms being utilized for extensive variety of streamlining issues. The suggested method takes into account a number of factors when choosing high-quality web pages, including the length of the website's existence, backward connections, forward links, and others. Faustina and Santosh [4] manage an investigation of various strategies and example of content mining and the regions which has been impacted by satisfied mining. The web contains organized, unstructured, semi-organized, and media information. This review centers around how to apply content mining on the above information. It additionally calls attention to how web content mining can be used in web use mining. Kohrs et al. [12] focus on the use of cooperative sifting for client adjusted sites. In order to solve common issues in cooperative filtration system and to enhance performance, the paper demonstrates ways for unifying and coordinating content-based filtering. Lee et al. [13] create an algorithm to precisely calculate user operation efficiency and make recommendations on how to improve it. There are two methods to do this: (i) by providing a new connection between two web sites, or (ii) by asking designers to rethink any existing ineffective links so that visitors may reach their destination pages more quickly. The author creates a prototype using this method to demonstrate the idea of efficiency. An adapting website system has been put into place to adjust the design of

the website automatically based on user browsing behavior and to increase website usability from an efficiency standpoint. In [1], a unique technique that concentrates on web archives by using information about web information, use, and design has been presented. The suggested method has applicability in a few key areas like web personalization, versatile site advancement, suggestion frameworks, website streamlining, business knowledge arrangements, and so on. Additionally, the suggested methodology has undergone preliminary analysis using the techniques WDPGA, WDPSA, and WDPPII. It has been determined that, with a slight time sacrifice, it has an advantage over these methodologies. Kumar and Kumar [14] propose choosing the top-K views for materialization, a randomized search heuristic algorithm based on modification and improvement is suggested. It is demonstrated that the suggested approach is capable of choosing comparably good quality views for greater dimensional datasets when compared to a well-known greedy algorithm. In [15], PSO algorithm for data mining has been proposed. PSO algorithm is population-based developmental heuristic inquiry techniques utilized for taking care of various combinatorial issues. In everyday, the standard created by affiliation rule mining strategy don't consider the negative events of credits in them, however, by utilizing PSO algorithm over these standards, the framework can foresee the principles which contains negative credits.

### 3 Prerequisite and Proposed Method

The proposed method for mining online information makes use of a different GA features. Darwin's theory of living things, claiming that fertile organisms were produced as a consequence of evolution, served as the inspiration for a subset of artificial intelligence known as GA. In other words, GA is a search approach that relies on genetics and on natural selection. The principal meaning of GA is natural selection which is otherwise called regular determination. It differs from earlier search techniques that look among populations of foci and employ boundary coding set instead of boundary values themselves. There are certain things that we can't determine without taking a series of actions to suggest a solution. The greatest tactic for such problems is search. There are two methods for doing searches. These heuristic methods and processes are for those who are blind. Blind processes don't make use of problem-space knowledge. Additional data is used via heuristic techniques to direct the search. Search system's two main objectives are exploring the search space and utilizing the optimum configuration. Utilizing existing sources is done via exploitation, while seeking for new ones is done through investigation. Hill climbing is a metaphor for exploitation, whereas random search is a metaphor for research. A wonderful balance is struck by GA between searching and manipulating the search space. The age of the population under consideration, considering the objective capacity, and using genetic regulators like generation, hybrid, and change is the main GA stages. GA execution begins with a populace of chromosomes which are arbitrarily created.

As indicated by the wellness capability, the chromosomes are evaluated. The chromosomes with improved arrangement are allowed greater opportunity to repeat than the chromosomes with more unfortunate arrangement. The suggested technique has been compared to the algorithm presented in future known as the MA algorithm. This work presents a GA-based strategy for online content mining to get the top-K web joins. Experimental evidence suggests that the suggested strategy outperforms the MA algorithm.

### ***3.1 Genetic Algorithm***

Genetic algorithm was first proposed by John Holland in 1960 at the College of Michigan. GA is evolutionary algorithm where evolution is a method for selecting a solution from a vast array of possible possibilities. Huge arrangement of possibility in biology refers to a number of potential genomic clusters. Finding the most fit living forms is the answer. Development may also be understood as a method for coming up with original solutions to challenging problems. Darwin's theory of evolution served as GA's driving force. Optimization challenges like applied mathematics and combinatorial optimization are handled by GA. Finding the ideal layout is the problem of optimization. The phrase 'best solution' implies that there are other configurations, each of which is not equally valuable. A GA mimics how development often happens. Initial population generation, evaluation of goodness of individuals in the initial population using some fitness function, selection of population for cross over, performing crossover, and mutation are the five phases. The biological foundation of GA stipulates that all living things are made up of cells. Furthermore, as chromosomes make up cells, each cell has a specific configuration of one or more chromosomes. They are made up of DNA strings that function as the biological entity's blueprint. Genes are also divided up into chromosomes. The genes are in charge of determining certain characteristics, such as eye color. Alleles are different scenarios in which a trait might exist. Eye color allele models include brown, blue, and hazel colors. Every gene's location on a chromosome is referred to as its locus. The entire configuration of genetic information is referred to as the organism genome. Genotype refers to a genome's particular gene organization. It is understood how a genotype is really expressed. The actual articulation of genotype is known as the phenotype.

### ***3.2 Particle Swarm Optimization***

PSO [15] is an iterative method that draws inspiration from bird flocks or fish schools that move in unison. In PSO, the population is referred to as a swarm, while the individuals are known as particles. Every particle looks for the optimal point, and this relies on the mobility and intelligence of the particle. Each particle movement

is then used to determine the particle's current location, best location, and additional best location. The fitness function evaluates the particle's present position.

PSO gives a significant undeniable level pieces of information for the underlying determination for additional grouping. The positioning and rate of adjustment of atoms or potential solutions in d-dimensional space are addressed. PSO encodes different layouts as a swarm of particles in the search space. A particle's initial values are selected at random. Starting at the beginning of the iteration, each molecule maintains track of the best results it has ever obtained. Every particle also has a certain neighborhood. Particles base their decisions on how well they and their neighbors are performing.

### 3.3 Problem Representation

Usually, any professional or commercial website has hundreds to thousands or even millions of pages, and each of these web pages has a corresponding URL. It would be cumbersome to process such a long URLs so these have been given a unique IDs as shown in Table 1. Further, from these, 'n' web pages candidate solutions are generated randomly as shown in Table 2. For large values of 'n', the number of possible candidate solutions would be very large as given in Eq. (1). At the same time, it would be computationally infeasible. The size of candidate solution is 'K' which is number of web documents to be ranked.

$$\text{Possible Candidate Solutions i.e. Size of Search Space} = C_t^n = \frac{n!}{t! \times (n-t)!} \quad (1)$$

where  $n$  is possible URLs and is large

$K$  is top-ranked ' $K$ ' web documents

**Table 1** Mapping of URLs with unique IDs

Webpages	WebP-1	WebP-2	WebP-3	...	WebP-n
IDs	1	2	3	...	$n$

**Table 2** Candidate solution

---

Sample one candidate solution of Top-5 web pages

---

Unique ids representing URLs generated randomly, Random(1,  $n$ )

45	32	97	147	2
----	----	----	-----	---

### 3.4 Cost Function

Cost function depends on various parameters described in Table 3.

$$\text{Cost}_{ac} = c1 \cdot \sum_i^t AF_i \quad (2)$$

$$\text{Cost}_{mv} = c2 \cdot \sum_i^t V_i \quad (3)$$

$$\text{Cost}_d = c3 \cdot \sum_i^t d_i \quad (4)$$

$$\text{Cost}_h = c4 \cdot \sum_i^t h_i \quad (5)$$

$$\text{Cost}_a = c5 \cdot \sum_i^t a_i \quad (6)$$

$$\text{Cost}_{kw} = c6 \cdot \sum_i^t kw_i \quad (7)$$

Using Eqs. (2) to (7), cost function can be formulated as given in Eq. (8). This cost function would be used to evaluate the goodness of candidate solution in proposed hybrid GA-PSO-based algorithm.

$$\text{Cost}(\text{Web } P_i) = \text{Cost}_{ac} + \text{Cost}_{mv} + \text{Cost}_d + \text{Cost}_h + \text{Cost}_a + \text{Cost}_{kw} \quad (8)$$

where  $c1, c2, c3, c4, c5$ , and  $c6$  are normalization constants.

**Table 3** Fitness function parameters

Parameters	Description
MAC	Max access count
MV	Max visitors
MD	Max duration spent by visitor on a web page
MH	Max hub value
MA	Max authority value
MK	Maximum keyword in a web page
KF	Frequency of keywords

### 3.5 Parameter Setting for PSO Algorithm

The basic PSO is given in algorithm 1. The algorithm starts by taking initial population and then evaluates the each particle using fitness function. Further, each set of top-K web document is updated based on the cost value. Better set of top-K web documents are retained, and thereafter, position and velocity of each candidate are updated. This procedure is repeated until the termination condition is met.

#### Algorithm 1

##### Abbreviations

f: Cost function, Vi: Cost of top-K web documents, A: Population of top-K webpages, W: Inertia weight, c1, c2: cognitive and social constant respectively, U<sub>1</sub>, U<sub>2</sub>: rand (0,1), Xi: Position of top-K web documents, P<sub>b</sub>: Best cost of each top-K web documents, g<sub>b</sub>: Globally best cost

##### Method

1. Generate a population of top-K (particles) documents randomly
2. Initialize the cost of each top-K web documents to a minimum value (large -ve number)
3. Gen = 1
4. Evaluate each set of top-K web documents using cost function  $Cost(WebP_i)$  using Eq. (8)
5. If cost (top-K web Docs) > cost (previous top-K web Docs)  
Find the set of best top-K web document
6. Update the cost of top-K web document using following equation

$$V_i^{t+1} = w \cdot V_i^t + c1 \cdot U_1^{t+1} (P_{bi}^t - P_i^t) + c2 \cdot U_2^t (g_b^t - P_i^t) \quad (9)$$

Move particles to their next positions

$$P_i^{t+1} = P_i^t + v_i^{t+1} \quad (10)$$

7. Gen = Gen + 1
8. if (Gen < Max\_Gen) go to step 4

If  $w = 1$ , the cost of current top-K web documents is influenced from cost of the previous top-K web documents so we will get the new set of top-K web documents around the previous population, because we are updating the cost value using Eqs. 9 and 10 and generating the new population around the updated cost by generating the different neighbors. If  $0 \leq w < 1$ , then we discard the top-K web documents because these are not the good solutions.  $P_{bi}^t$  is best cost of an individual, and  $P_i^t$  is its current cost (i.e., position). It is observed that as the top-K web document (particle) is away

in cost from  $P_{bi}^t$  (personal best), the value  $(P_{bi}^t - P_i^t)$  will increase, and thus, term increases, attracting the particle to its best own position.  $c1$  is a positive constant. It signifies the importance of top-K document's previous experiences.

$U_1^t$  random number between [0,1] which stops the premature convergence. The value  $(g_b^t - P_i^t)$  is used to find best solution.  $c2$  is social learning parameter and denotes the global learning of particles.  $U_2^t$  is similar to  $U_1^t$ .

*If  $c1 = c2 = 0$ , top-K web documents will come from the search space.*

*If  $c1 > 0$  and  $c2 = 0$ , all top-K web documents are independent.*

*If  $c1 > 0$  and  $c2 = 0$ , all top-K web documents will converge to single solution.*

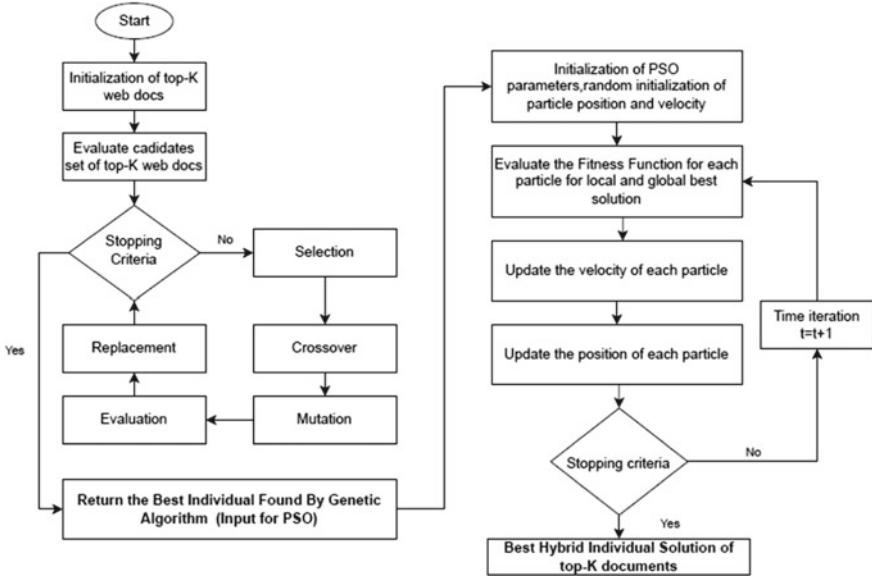
*If  $c1 = c2 \neq 0$ , all top-K web documents will give the average value of  $p_{best}$  and  $g_{best}$ .*

### 3.6 The Proposed Hybrid Approach Based on GA and PSO

The flowchart of the proposed hybrid approach is given in Fig. 1, where first the genetic algorithm is applied to the set of initial top-K candidate solutions generated randomly from the total set of web pages available in the website. For this, each web page is given a unique numerical ID as shown in Table 1. Then, set of top-K candidate solutions are generated out of these numerical IDs. Next each candidate solution is evaluated for its goodness using the cost function given in Eq. 8. Thereafter, the basic GA operators, selection, crossover, mutation, evaluation, and replacement, are applied until the stopping criteria are not met. Whenever, stopping criteria are met, the optimal candidate solution is returned, and it becomes input for the PSO algorithm. The PSO starts by taking different initial population, and GA is applied several times to get enough number of particles for PSO-based algorithm. Due to this, PSO-based algorithm would get the improved initial population. The PSO starts by initializing its parameters, particle position, and velocity. Next, each particle is evaluated for its fitness, and local and global best solution is found. Then, updation of position and velocity of each particle takes place. Here, the particles are considered as set of top-K web documents. The procedure is repeated until the termination condition is found, and finally, the most relevant top-K web document is returned as result.

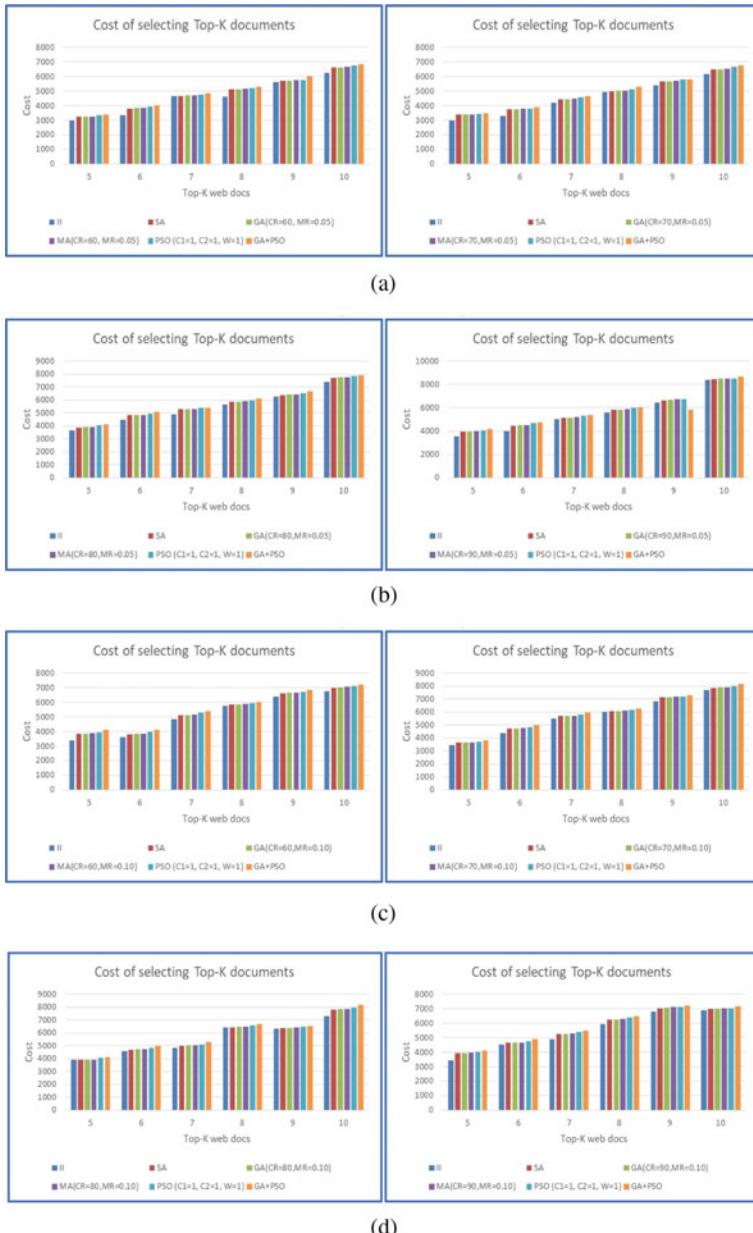
## 4 Result and Discussion

The web documents algorithms based on iterative improvement (II) [2], simulated annealing (SA) [9], GA [5, 8, 9], memetic algorithms [1], and PSO have been implemented in Java on windows platform. The GA has been run for different combinations of crossover rate of (60, 70, 80, 90) and mutation rate of (5%, 10%). The II-based approach has been tested for different number of neighbors, 30, 40, 50, and 60, and it has been found that for 60 number of neighbors, II is generating better quality of top-K web documents. The SA-based approach has been run by setting the initial



**Fig. 1** Proposed hybrid GA + PSO approach for selecting top-K web documents

temperature value which is the two times the cost of initial random state (candidate solution). The GA and MA-based approaches have been run for different combination of crossover (60, 70, 80, 90) and mutation rate (0.05, 0.10). Results are shown in Fig. 2. It can be seen from the Fig. 2 that hybrid, GA-PSO approach is able to select better quality top-ranked web documents. PSO has been run for  $c_1 = 1$ ,  $c_2 = 1$ , and  $W = 1$ . It can also be seen from the graph that the hybrid approach is scalable for higher dimensions for selecting Top-K documents and is able to select better ranked web documents in comparison with all other approaches.



**Fig. 2** Comparative analysis of selection of top-K web documents by hybrid GA + PSO approach  
**a** GA[CR = 60, MR = 0.05] versus PSO[c1 = c2 = 1, w = 1] **b** GA[CR = 70, MR = 0.05] versus PSO[c1 = c2 = 1, w = 1] **c** GA[CR = 80, MR = 0.05] versus PSO[c1 = c2 = 1, w = 1] **d** GA[CR = 90, MR = 0.05] versus PSO[c1 = c2 = 1, w = 1] **e** GA[CR = 60, MR = 0.10] versus PSO[c1 = c2 = 1, w = 1] **f** GA[CR = 70, MR = 0.10] versus PSO[c1 = c2 = 1, w = 1] **g** GA[CR = 80, MR = 0.10] versus PSO[c1 = c2 = 1, w = 1] **h** GA[CR = 90, MR = 0.10] versus PSO[c1 = c2 = 1, w = 1]

## 5 Conclusion

In this paper, a hybrid GA-PSO-based approach for selecting top-K web documents has been proposed. In this, GA has been applied on the initial population of size 100 which has been generated randomly, and the optimized set of top-K web documents have been further optimized by the PSO algorithms. The GA algorithm is run several times to generate enough number of particles for the PSO algorithm. Many relevant features based on web usage, structure, and content data have been used to rank the web pages for reorganizing the structure of website. These features are keyword frequency, hubs and authority values, active time spend by the user on any web page, number of unique visitors, and access account. The ranking function uses all these features to prioritize web pages. This ranking function makes the proposed approach a novel contribution in the literature. Experimental results show that the hybrid approach is able to select better quality top-K web documents as selected by individual algorithms based on II, SA, GA, and PSO.

## References

1. Kumar S, Kumar R (2021) WDPMA: an MA-based model for web documents prioritization. *Int J Inf Technol Web Eng* 16:1–24. <https://doi.org/10.4018/IJITWE.2021040101>
2. Chaudhary K, Gupta SK (2014) Prioritizing web links based on web usage and content data. In: 2014 International conference on issues and challenges in intelligent computing techniques (ICICT), pp 546–551. <https://doi.org/10.1109/ICICICT.2014.6781340>
3. Dogan A, Birant D (2021) Machine learning and data mining in manufacturing. *Expert Syst Appl* 166:114060
4. Johnson F, Gupta S (2012) Web content mining techniques: a survey. *Int J Comput Appl* 47:44–50. <https://doi.org/10.5120/7236-0266>
5. Johnson F, Kumar S (2013) Web content mining using genetic algorithm. In: Unnikrishnan S, Surve S, Bhoir D (eds) Advances in computing, communication, and control. ICAC3 2013. Communications in computer and information science, vol 361. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-36321-4\\_8](https://doi.org/10.1007/978-3-642-36321-4_8)
6. Kumar S, Kumar R (2021) A study on different aspects of web mining and research issues. In: IOP conference series: materials science and engineering. vol 1022. pp 012018. <https://doi.org/10.1088/1757-899X/1022/1/012018>
7. Irfan S, Dhanaraj RK (2021) BeeRank: a heuristic ranking model to optimize the retrieval process. *Int J Swarm Intell Res (IJSIR)* 12(2):39–56. [https://doi.org/10.4018/IJSIR.202104\\_0103](https://doi.org/10.4018/IJSIR.202104_0103)
8. Gupta SK, Singh D, Doegar A (2016) Web documents prioritization using genetic algorithm. In: 2016 3rd International conference on computing for sustainable global development (INDIACom), pp 3042–3047
9. Shenoy P, Srivatsa K, Thomas A, Venugopal KR, Lalit P (2005) Mining top—k ranked webpages using simulated annealing and genetic algorithms. vol 3285. pp 137–144. [https://doi.org/10.1007/978-3-540-30176-9\\_18](https://doi.org/10.1007/978-3-540-30176-9_18)
10. Lee J-H, Shiu W-K (2004) An adaptive website system to improve efficiency with ant colony algorithm. *Adv Eng Inform* 18:129–142. <https://doi.org/10.1016/j.aei.2004.09.007>
11. Marghny MH, Ali AF (2005) Web mining based on genetic algorithm. In AIML 05 Conference, December, pp 82–87

12. Kohrs A, Merialdo B (2001) Creating user-adapted websites by the use of collaborative filtering. *Interact Comput* 13:695–716. [https://doi.org/10.1016/S0953-5438\(01\)00038-8](https://doi.org/10.1016/S0953-5438(01)00038-8)
13. Lee J-H, Shiu W-K (2004) An adaptive website system to improve efficiency with web mining techniques. *Adv Eng Inform* 18:129–142. <https://doi.org/10.1016/j.aei.2004.09.007>
14. Kumar T, Kumar S (2013) Materialized view selection using iterative improvement. [https://doi.org/10.1007/978-3-642-31600-5\\_21](https://doi.org/10.1007/978-3-642-31600-5_21)
15. Su T, Xu H, Zhou X (2019) Particle swarm optimization-based association rule mining in big data environment. *IEEE Access* 1–1. <https://doi.org/10.1109/ACCESS.2019.2951195>

# Addressing Class Imbalance in Fake News Detection with Latent Space Resampling



Saranya Bhattacharjee, Soumyajit Maity, and Sankhadeep Chatterjee

**Abstract** The detection of fake news has become crucial with the popularity of social media as a primary medium of news consumption. However, real-world fake news datasets often suffer from the problem of class imbalance. These imbalanced datasets lead to unreliable and skewed classifier performance. Motivated by this, our paper addresses the class imbalance problem in fake news detection by employing a deep learning-based latent space resampling strategy. Firstly, a Bidirectional Variational Autoencoder is utilized to attain the most informative latent representations of the input text sequences. Next, the class-biased latent vectors are resampled using oversampling, undersampling, and hybrid-sampling techniques. The resampled latent vectors are then used to train several state-of-the-art classification algorithms and evaluated using a separate balanced testing set. Performance comparison with the imbalanced set serving as the baseline model and state-of-the-art methods, on the basis of the F1-score and Geometric mean, reveals that the proposed architecture is capable of alleviating the problem of class bias considerably.

**Keywords** Fake news detection · Social media · Class imbalance · Resampling

## 1 Introduction

News consumption on social media has skyrocketed in the last few years. Owing to its rapid dissemination and accessibility, people prefer to browse through online social networks rather than relying on traditional news sources [1]. However, this comes at the expense of being exposed to “fake news.” For a number of reasons, including financial and political benefits, massive volumes of fake news (news con-

---

S. Bhattacharjee · S. Maity

Department of Computer Science and Engineering, University of Engineering and Management, Kolkata 700160, India

S. Chatterjee (✉)

Department of Computer Science and Technology, University of Engineering and Management, Kolkata 700160, India

e-mail: [chatterjeesankhadeep.cu@gmail.com](mailto:chatterjeesankhadeep.cu@gmail.com)

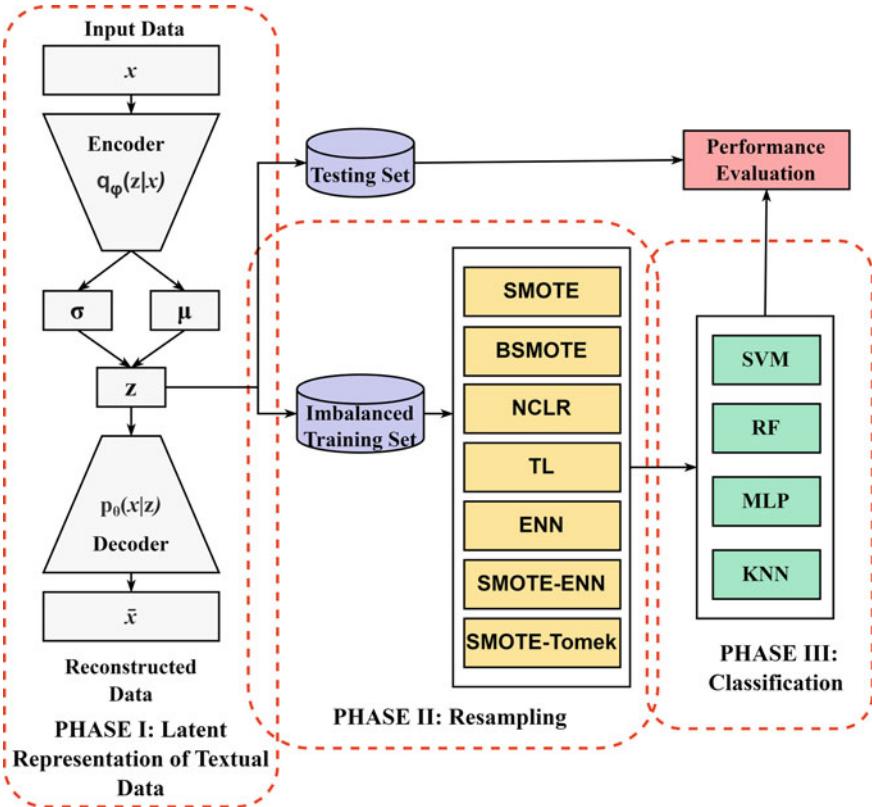
taining false and misleading information) are created online [2]. Hence, identifying such false news is a crucial task that not only guarantees users obtain accurate information but also supports the preservation of a reliable news ecosystem. There have been several studies on fake news detection that have been investigated in the recent past [3, 4]. However, in real-world fake news datasets, the number of fake news data samples is generally less than the number of real news data samples. This imbalanced distribution results in a biased classifier performance. In [5], the authors use oversampling strategies to deal with class bias in textual data. Yet, other undersampling and hybrid-sampling techniques are not studied. Moreover, traditional data augmentation approaches fail to obtain representative features and mitigate the class imbalance problem by incorporating Natural Language Processing (NLP) techniques. In one such instance, the authors in [6] proposed an ensemble classification model to detect fake news. The ensemble approach recorded a poor test accuracy of 44.15% on the ISOT dataset. Consequently, to address these issues, the present study proposes a deep learning-based latent space resampling strategy to mitigate the effects of class imbalance observed in fake news detection. Furthermore, the influence of three different sampling strategies, viz., oversampling, undersampling, and hybrid-sampling methods on the detection of fake news is also explored. Overall, the contributions of the paper are as follows:

1. In order to provide an appropriate latent vector representation of the input text sequences, a BiLSTM Variational Autoencoder architecture is developed.
2. A variety of oversampling, undersampling, and hybrid-sampling methods are employed to balance the training set.
3. Four traditional classifiers are trained and evaluated using a balanced testing technique.

The rest of the paper is arranged as follows: The literature survey of the paper is presented in Sect. 2. Sections 3 and 4 describe the proposed approach and experimental setup respectively. Further, Sect. 5 elaborates on the results obtained by our experiments. Finally, Sect. 6 concludes the paper and discusses the future scope of the study.

## 2 Literature Survey

Prior studies have indicated the importance of fake news detection in today's day and age. In [7], the authors shed light on identifying fake news in digital media. COVID-19 tweet data was used to analyze and detect fake news in [8]. In [9], a hybrid-graph neural network-based methodology was employed to learn auxiliary data based on social settings to detect false information. However, class imbalance in fake news detection is one area where there has not been much focus. In [10], fake news detection in Bangla was investigated where the study centered around imbalanced datasets. Furthermore, multimodal detection of fake news was studied



**Fig. 1** Proposed methodology

in [11] with feature fusion and a Bidirectional LSTM network. In [12], a binary text classifier with the help of an MLP network was utilized for the detection of fake news articles. BERT-based fake news classification was demonstrated in [13].

### 3 Proposed Approach

In this section, we elaborate on the proposed architecture employed to mitigate the inherent problem of class bias in fake news detection. To do so, as shown in Fig. 1, we divide the process into three integral phases.

### 3.1 Latent Representation of Textual Data

The first phase of our proposed architecture involves training a Bidirectional LSTM Variational Autoencoder (BiLSTM VAE) with training data. In the last few years, there have been numerous studies on generative modeling that have demonstrated how the most relevant features of textual data can be used to improve classifier performance [5]. This is based on the premise that the latent vector will hold onto the most representative and important information from the original data provided the VAE is appropriately trained to reconstruct the data. Motivated by this, our approach uses a BiLSTM VAE to obtain a compact feature vector of the textual data. A batch contains padded sequences that are of the same length. The Embedding Layer accepts a 2D tensor and returns a 3D tensor. The embedding dimension has been set at 150 for our experiment. The BiLSTM layers in the encoder then process it. The memory unit size of the first BiLSTM layer is 128; as a result, each hidden state has 256 dimensions. The forward and backward passes are combined in the second BiLSTM layer's unit of size 64, producing an output shape of (None, 128). In this study, the latent vector of size 128 is used [5]. Further, the BiLSTM VAE [14] maps the data instances to a multivariate normal distribution, parameterized by mean  $\mu$  and log-variance  $\sigma$  vectors. The encoder network ( $q_\phi(z | x)$ ) compresses the input text sequence  $x$  to a latent vector  $z$  (via sampling), and then  $z$  is reconstructed back to a sequence  $\bar{x}$  similar to the original text data, by a decoder network ( $p_\theta(x | z)$ ). The objective function is described in Eq. (1).

$$L_{\text{VAE}}(\theta, \phi) = -\mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x | z) + D_{\text{KL}}(q_\phi(z | x) \| p_\theta(z)) \quad (1)$$

where the first term represents the reconstruction error and the second term denotes the regularizer, i.e., Kullback-Leibler divergence between the distribution of encoder  $q_\phi(z | x)$  and  $p(z)$ . In addition, the Reparameterization trick permits backpropagation to pass through the deterministic nodes by including the  $\epsilon$  parameter, which is achieved by Eq. (2):

$$\begin{aligned} \mathbf{z} &= \mu + \sigma \odot \epsilon \\ \epsilon &\sim \mathcal{N}(0, I) \end{aligned} \quad (2)$$

where  $\sigma$  and  $\mu$  exhibits the standard deviation and mean respectively, and  $\odot$  means element-wise product. Here, although the training set is imbalanced, the BiLSTM VAE training remains unaffected by class bias since it is an unsupervised training process. Once the training is complete, the individual training and testing sets are sent to the encoder, and the respective compressed feature vectors are stored with their class labels.

**Table 1** Illustrating data before and after resampling

Resampling technique	Number of samples			
	Before resampling		After resampling	
	0	1	0	1
SMOTE	9250	29207	29207	29207
BSMOTE	9250	29207	29207	29207
NCLR	9250	29207	9250	24538
ENN	9250	29207	9250	22029
TL	9250	29207	9250	28182
SMOTE-ENN	9250	29207	26681	20420
SMOTE-Tomek	9250	29207	29134	29134

### 3.2 Resampling

The second phase of the proposed architecture (see Fig. 1) is the Resampling phase. The imbalanced training set, here, is 100% resampled using 2 oversampling techniques, viz., SMOTE [15] and BorderlineSMOTE (BSMOTE) [16], 3 undersampling techniques, viz., Neighborhood Cleaning Rule (NCLR) [17], Edited Nearest Neighbors (ENN) [18], Tomek Links (TL) [19], and 2 hybrid-sampling techniques, viz. SMOTE-ENN [20] and SMOTE-Tomek [20].

Oversampling strategies build balanced datasets by replicating existing data samples or by creating new samples. Synthetic Minority Oversampling TEchnique (SMOTE) is one of the first resampling techniques to overpopulate the minority class instances. The process follows iterating through each of the minority data samples and implanting a synthetic data point along the line joining a distinct minority sample and its  $k$  nearest minority samples. BSMOTE is a reworked variant of the standard SMOTE aimed to populate just the minority samples found in the class's boundary region.

On the other hand, undersampling techniques eliminate majority class data instances to solve for class bias. TL identifies noisy majority samples that are forming tomek links with the minority class and are removed. NCLR and ENN undersampling techniques are based on the element of misclassification. Further, hybrid-sampling methods combine various sampling techniques to minimize the drawbacks of each methodology alone. SMOTE-ENN and SMOTE-Tomek combine the algorithms of oversampling and undersampling where ENN and Tomek Links perform data cleaning to the synthetic examples generated by SMOTE, respectively.

Table 1 exhibits the training data before and after resampling where '0' denotes fake news data and '1' denotes real news data.

### 3.3 Classification

In the classification stage, 4 traditional classification algorithms, viz., Random Forest (RF),  $k$ -Nearest Neighbor (KNN) classifier, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP) have been implemented. The resampled training set is used to train the aforementioned classifiers, which are then evaluated based on the performance metrics using the balanced testing set.

## 4 Experimental Setup

In this section, we discuss the dataset used in the current study. We also summarize the parametric setup of resampling algorithms and BiLSTM VAE model.

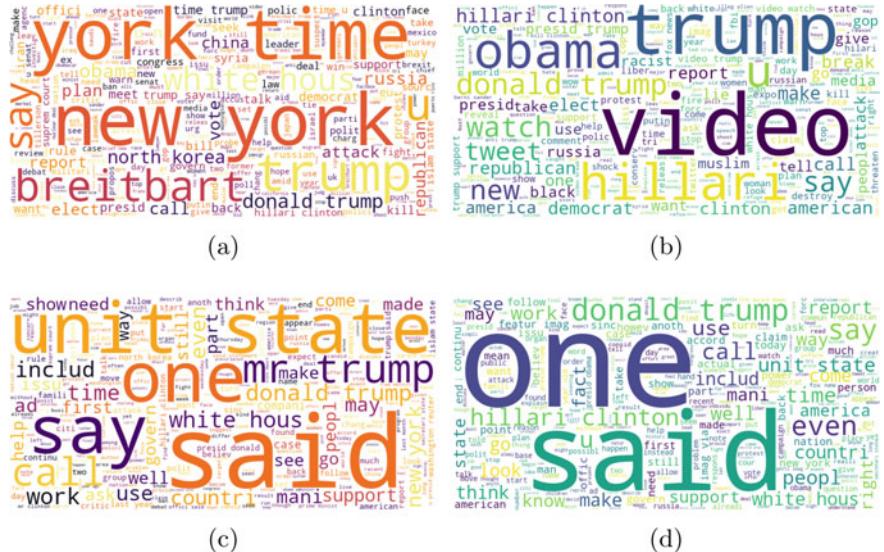
### 4.1 Dataset Overview

The WELFake Data Set [21] has been used to test our proposed methodology. The original dataset contains 72,134 news articles with 35,028 real and 37,106 fake news. For the purpose of minimizing classifier overfitting, the authors integrated four prominent news datasets (Kaggle, McIntire, Reuters, and BuzzFeed Political). However, as stated in recent literature [22], the biased class distribution (i.e., class imbalance) of the dataset is a predominant problem in fake news detection. In such cases, the number of fake news is generally less compared to real news data [23]. Hence, to replicate a real-world scenario, we randomly subsample the fake news class data samples. The newly formed training dataset comprises 29,207 real and 9250 fake news instances. The fake news class forms the minority class and, in our experiment, the class of interest. Meanwhile, to have an honest and reliable evaluation, the testing set is balanced having 5000 data samples in each class.

Figure 2 illustrates the Wordcloud visualizations of the dataset based on ‘fake titles’, ‘real titles’, ‘fake texts’, and ‘real texts’. Data cleaning and pre-processing have been conducted where all texts are converted to lowercase and whitespace, punctuations, and stopwords are removed.

### 4.2 Model Parameter Setup

In the case of the BiLSTM VAE training, a learning rate of  $1e-4$ , sparse categorical cross entropy as the loss function, and Nadam, as the model optimizer have been employed. The size of the batch is 100 and the number of epochs is 20. The default parameters have been implemented in the case of the resampling techniques. Addi-



**Fig. 2** Wordcloud visualization based on **a** Fake titles, **b** real titles, **c** fake texts, and **d** Real texts

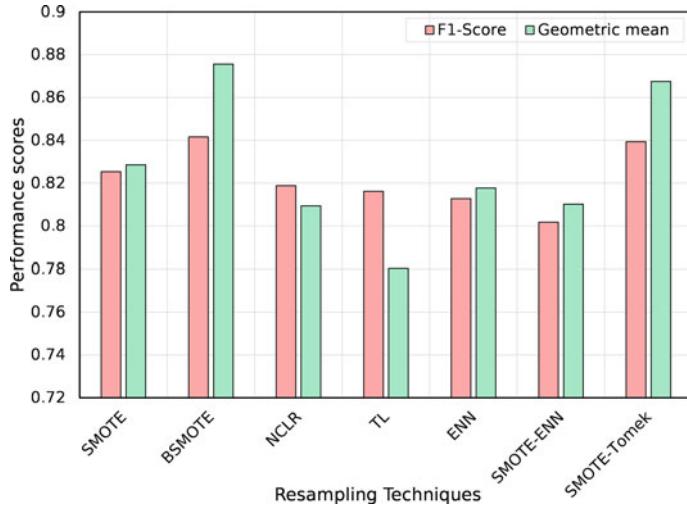
tionally, we have enforced Grid Search to determine the hyper-parameters of the classifier models. For the KNN classifier, the number of neighbors is chosen to be 10. In the case of RF, 100 trees are taken in the forest. At the same time, for the MLP classifier, the activation function set is ReLU, with Adam solver, the learning rate is designated a constant value of 0.001 and the maximum iteration is 1000. While for SVM, the regularization parameter is selected to be 1.0, with a tolerance of 0.001, and the probability estimates are enabled.

Additionally, to measure the efficacy of our proposed approach, we have analyzed our investigations with a baseline model. Here, the imbalanced training set is utilized to train the classification methods and trained models are tested using a testing set that is balanced. This method does not employ any resampling to balance the dataset's class distribution.

## 5 Results and Discussion

In this section, we assess our proposed architecture on the basis of two performance metrics, viz., F1-score and geometric mean. These metrics are expressed as follows:

$$\text{F1-Score} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}} \quad (3)$$



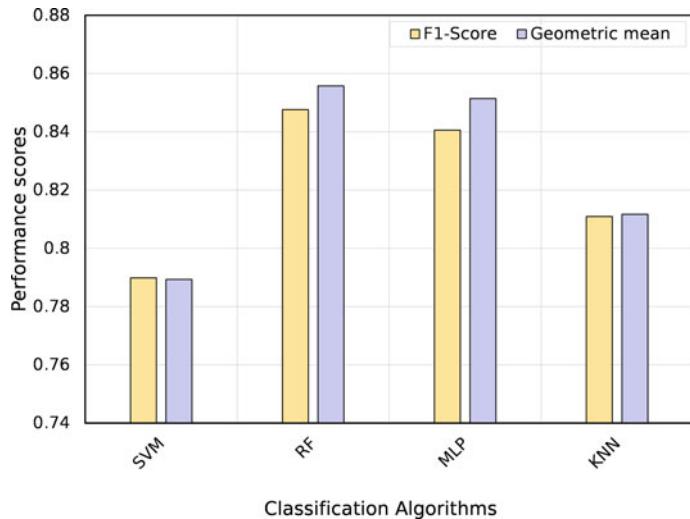
**Fig. 3** Performance comparison of resampling techniques on the basis of F1-score and geometric mean

$$\text{Geometric mean} = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \cdot \frac{\text{TN}}{\text{TN} + \text{FP}}} \quad (4)$$

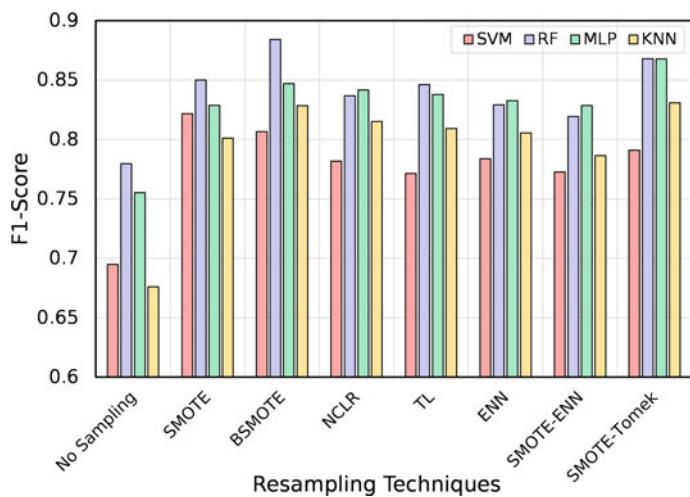
where, TN, FP, TP, and FN denote true negative, false positive, true positive, and false negative, respectively. As observed in recent literature [5], these evaluation metrics are able to capture the classifier's performance gain of the minority data samples which are mostly overlooked in case of extreme class imbalance.

Figures 3 and 4 demonstrate the scores with regards to F1-score and geometric mean of each of the resampling and classification techniques, respectively. In the case of Fig. 3, we have considered the average of all the individual performance scores generated by the classifiers using a specific resampling approach in order to examine the most effective resampling methodology. Similarly as observed in Fig. 4, the average of the corresponding metric values of all resampling strategies for a specific classifier has been determined in order to identify the classification method that accomplishes the optimal results under different sampling methods. According to Figs. 3 and 4, BSMOTE oversampling technique and RF classifier have been the best-performing resampling technique and classifier, respectively. The performance of the SMOTE-Tomek hybrid-sampling strategy and MLP classifier have also been very promising.

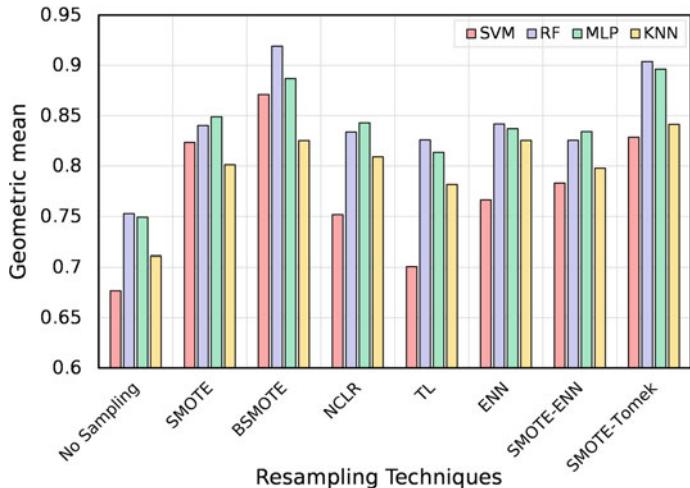
Figures 5 and 6 illustrate the performance of the 4 classifiers with 7 different resampling techniques on the basis of F1-score and geometric mean. The two respective plots depict the scores (as represented by the y-axis) w.r.t. a performance metric obtained by the resampling integrated with classification methods (as represented by the x-axis). Here, the initial imbalanced labeled set is used to train each of the classifiers (expressed as No Sampling + SVM, No Sampling + RF, No Sampling + MLP,



**Fig. 4** Performance comparison of classification algorithms on the basis of F1-score and geometric mean



**Fig. 5** Performance evaluation based on F1-score



**Fig. 6** Performance evaluation based on geometric mean

**Table 2** Performance comparison with state-of-the-art methods

Method	F1-score	Geometric mean
[12]	0.82	—
[13]	0.8376	—
Proposed method	0.8841	0.9191

and No Sampling + KNN) which serves as the baseline for comparison. It can be observed that after resampling the imbalanced labeled set, there has been a significant performance improvement. RF + BSMOTE records the highest F1-score and geometric mean of 0.8841 and 0.9191, respectively. Furthermore, MLP + SMOTE-Tomek documents a noticeable geometric mean score of 0.8962. Table 2 demonstrates the comparative assessment of our method with recent state-of-the-art techniques.

## 6 Conclusion

The current study proposed a latent space resampling strategy to alleviate the problem of class imbalance in fake news detection. Firstly, a BiLSTM VAE is trained to learn the most important latent feature vectors of the input text sequences. On obtaining the imbalanced training latent vectors, we have employed 7 resampling techniques, viz., SMOTE, BSMOTE, NCLR, ENN, TL, SMOTE-ENN, and SMOTE-Tomek, to balance the class distribution. Next, 4 state-of-the-art classification algorithms have been adopted to classify real and fake news and tested with the help of a balanced

testing set. Experimental results convey that RF + BSMOTE has been the most optimal resampling and classifier combination to attain the highest performances with respect to F1-score and geometric mean. In addition, performance comparison with the baseline model demonstrates that our proposed approach is capable of mitigating the data imbalance problem by focusing on the minority class samples.

Future research can focus on investigating generative adversarial networks in the context of class imbalance. Moreover, code-mixed social media data and multimodal datasets can also be studied in the future.

## References

1. Capuano N, Fenza G, Loia V, Nota FD (2023) Content based fake news detection with machine and deep learning: a systematic review. *Neurocomputing*
2. Shah MN, Ganatra A (2022) A systematic literature review and existing challenges toward fake news detection models. *Soc Netw Anal Mining* 12(1):168
3. Davoudi M, Moosavi MR, Sadreddini MH (2022) Dss: a hybrid deep model for fake news detection using propagation tree and stance network. *Exp Syst Appl* 198:116635
4. Hamed SK, Ab Aziz MJ, Yaakub MR (2023) Fake news detection model on social media by leveraging sentiment analysis of news content and emotion analysis of users' comments. *Sensors* 23(4):1748
5. Bhattacharjee S, Maity S, Sen R, Chatterjee S (2022) Class biased sarcasm detection using variational lstm autoencoder. In: Proceedings of international conference on computational intelligence, data science and cloud computing: IEM-ICDC 2021, pp 289–297. Springer, Berlin
6. Hakak S, Alazab M, Khan S, Gadekallu TR, Reddy Maddikunta PK, Khan WZ (2021) An ensemble machine learning approach through effective feature extraction to classify fake news. *Fut Gener Comput Syst* 117:47–58
7. Shahzad K, Khan SA, Ahmad S, Iqbal A (2022) A scoping review of the relationship of big data analytics with context-based fake news detection on digital media in data age. *Sustainability* 14(21):14365
8. Mehta V, Mishra RK (2022) Machine learning based fake news detection on covid-19 tweets data. In: Proceedings of international conference on computational intelligence and data engineering: ICCIDE 2021, pp 89–96. Springer, Berlin
9. Saikia P, Gundale K, Jain A, Jadeja D, Patel H, Roy M (2022) Modelling social context for fake news detection: a graph neural network based approach. In: 2022 international joint conference on neural networks (IJCNN), pp 01–08. IEEE
10. Hossain MdM, Awosaf Z, Hossan Prottoy MdS, Muhammod Alvy AS, Morol MdK (2022) Approaches for improving the performance of fake news detection in Bangla: imbalance handling and model stacking. In: Proceedings of international conference on fourth industrial revolution and beyond, 2021, pp 723–734. Springer, Berlin
11. Kishore V, Kumar M (2023) Enhanced multimodal fake news detection with optimal feature fusion and modified bi-lstm architecture. In: *Cybernetics and systems*, pp 1–31
12. Rusli A, Young JC, Iswari NiMS (2020) Identifying fake news in indonesian via supervised binary text classification. In: 2020 IEEE international conference on Industry 4.0, artificial intelligence, and communications technology (IAICT), pp 86–90. IEEE
13. Kumari S (2021) Nofake at checkthat! 2021: fake news detection using bert. arXiv preprint [arXiv:2108.05419](https://arxiv.org/abs/2108.05419)
14. Chatterjee S, Bhattacharjee S, Ghosh K, Das AK, Banerjee S (2023) Class-biased sarcasm detection using bilstm variational autoencoder-based synthetic oversampling. *Soft Comput* 1–18

15. Chawla NV, Bowyer KW, Hall LO, Philip Kegelmeyer W (2002) Smote: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
16. Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. International conference on intelligent computing. Springer, Berlin, pp 878–887
17. Laurikkala J (2001) Improving identification of difficult small classes by balancing class distribution. Conference on artificial intelligence in medicine in Europe. Springer, Berlin, pp 63–66
18. Bach M, Werner A, Żywiec J, Pluskiewicz W (2017) The study of under-and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis. *Inform Sci* 384:174–190
19. Debashree D, Biswajit P et al (2017) Redundancy-driven modified Tomek-link based undersampling: a solution to class imbalance. *Pattern Recogn Lett* 93:3–12
20. Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor Newslett* 6(1):20–29
21. Verma PK, Agrawal P, Amorim I, Prodan R (2021) Welfare: word embedding over linguistic features for fake news detection. *IEEE Trans Comput Soc Syst* 8(4):881–893
22. Kang M, Seo J, Park C, Lim H (2022) Utilization strategy of user engagements in korean fake news detection. *IEEE Access* 10:79516–79525
23. Wang W, Wang S, Fan W, Liu Z, Tang J (2020) Global-and-local aware data generation for the class imbalance problem. In: Proceedings of the 2020 SIAM international conference on data mining. SIAM, pp 307–315

# MEDNet-Based Imbalanced Cataract Detection Using Ophthalmic Images



Soumyajit Maity, Saranya Bhattacharjee, Ankur Das,  
and Sankhadeep Chatterjee

**Abstract** Cataract is a common age-related eye disease that causes cloudy vision and can lead to blindness if left untreated. Early detection of cataracts is crucial for timely treatment and to prevent vision loss. While there are established methods for cataract detection, they often struggle with imbalanced datasets. This work aims to improve the accuracy and speed of cataract detection using a MEDNet-based model and proper data preprocessing techniques. The model was trained on a dataset of imbalanced eye images that were first converted into a latent vector form using MEDNet. Then, popular sampling methods were used to balance the dataset. Finally, the model was evaluated using cross-validation and compared to existing models for accuracy and speed. The results of our experiment show that the MEDNet-based model improved the accuracy of cataract detection compared to existing methods. The use of latent vectors and sampling techniques helped to balance the dataset and improve the model's performance on imbalanced data. This work is an important step toward accurate and early detection of cataracts, which can improve patient outcomes and prevent blindness. The proposed MEDNet-based cataract detection model has the potential to be used as a reliable and automated tool for the early detection of cataracts from ophthalmic images, which could help reduce the burden on ophthalmologists and improve the quality of eye care.

**Keywords** Cataract detection · Ophthalmic images · Imbalance · Resampling

---

S. Maity · S. Bhattacharjee

Department of Computer Science and Engineering, University of Engineering & Management, Kolkata, West Bengal 700160, India

A. Das

Department of Software Engineering, Concordia University, Montreal H3G 1M8, Canada

S. Chatterjee (✉)

Department of Computer Science and Technology, University of Engineering & Management, Kolkata, West Bengal 700160, India

e-mail: [chatterjeesankhadeep.cu@gmail.com](mailto:chatterjeesankhadeep.cu@gmail.com)

## 1 Introduction

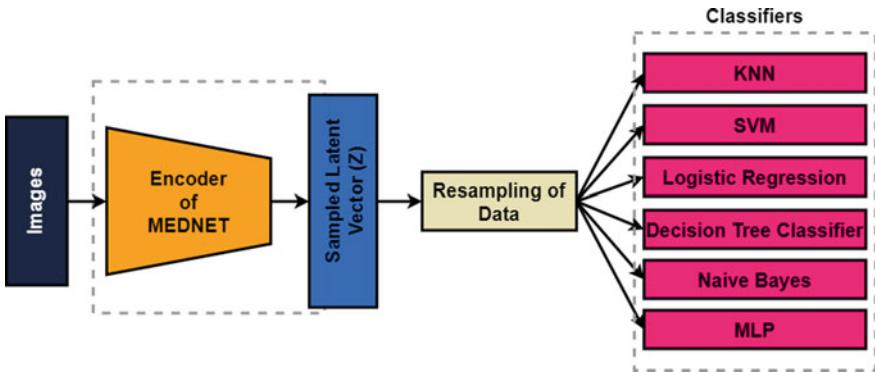
Cataracts, a widespread ocular condition, impact a large number of people globally, with developing countries experiencing a significant burden. Detecting cataracts in their early stages and providing prompt treatment is critical for preventing vision loss and enhancing the well-being of those affected. The hallmark of cataracts is the opacification or clouding of the eye's lens, which results in a gradual decline in visual acuity. In the USA, cataract affects more than 24 million people aged 40 and over [1]. Globally, it is estimated that 65 million people have cataracts, accounting for 33% of all causes of blindness [2].

Automated cataract detection from digital images has the potential to assist ophthalmologists in the early detection and diagnosis of cataracts and help improve the quality and efficiency of eye care. Digital imaging technology has enabled the study of biomedical images for machine learning and digital image processing. Eye imaging techniques, including optical coherence tomography (OCT) and slit-lamp biomicroscopy, are commonly used to capture digital images of the eye, which can be analyzed for automated cataract detection. In this work, we propose a MEDNet-based [3] cataract detection model that effectively addresses the imbalance problem and improves the performance of automated cataract detection from digital images. Our proposed method achieved exceptional results, with an accuracy, recall, F1-score of 0.99 and precision score of 1 surpassing current industry standards. The high scores obtained by our method indicate its effectiveness in accurately classifying the data and its potential to outperform existing approaches in various applications.

## 2 Literature Survey

The detection of cataracts from ophthalmic images is a highly active area of research due to its prevalence as the leading cause of blindness worldwide. Researchers have proposed several methods for detecting cataracts from ophthalmic images [4], including machine learning techniques [5], image processing [6], and deep learning-based methods [7]. Texture analysis involves extracting features based on texture patterns, while convolutional neural networks (CNNs) can learn features directly from images without manual feature extraction [8]. Furthermore, image enhancement techniques such as adjusting contrast and brightness have been explored to improve the accuracy of cataract detection [9].

A major challenge in cataract detection using ophthalmic images is the class imbalance problem, where the cataract images class is less represented in the dataset as compared to the non-cataract class, thereby leading to biased results. To address this issue, researchers have proposed several methods, including oversampling, undersampling, generating synthetic images using Generative adversarial networks (GANs) [10] and using cost-sensitive learning [11]. In [12], the authors present the DKCNet model along with resampling methods to enhance the classification of fun-

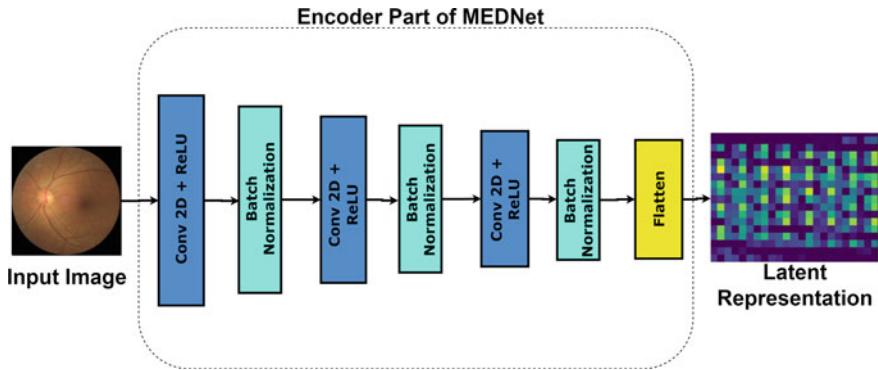


**Fig. 1** MEDNet is fed with ophthalmic images and obtained latent vectors are resampled. Then, the classification algorithms are trained using the balanced set and evaluated

dus images. Jiang et al. [13] introduced a novel approach for identifying ophthalmic disorders using retro-illumination images. The authors developed a cost-sensitive deep residual convolutional neural network (CS-ResCNN) classifier that identifies the regions of interest, particularly the crystalline lens, to overcome the challenge of dataset imbalance. Similarly, Kim et al. [14] proposed a new architecture for convolutional neural networks, called Tournament-based Ranking CNN, which addresses the dataset imbalance issue by enhancing the Ranking CNN method that combines the results from multiple binary neural network models.

### 3 Proposed Method

This section outlines the methodology we employed to detect cataracts using the MEDNet model on an imbalanced dataset. Our methodology comprises the following steps: (1) The original imbalanced dataset is processed by the MEDNet model to generate the latent vector. (2) To mitigate the problem of class imbalance, we utilized different well-known resampling techniques to resample the data. (3) Finally, the performance of the classifiers is measured using appropriate performance metrics. This methodology aims to achieve accurate and efficient detection of cataracts, which is crucial in identifying the disease in its early stages and initiating prompt treatment. The use of the MEDNet model and resampling techniques are expected to improve the accuracy of the classification, especially in the case of an imbalanced dataset. The performance metrics used in this study will provide insights into the effectiveness of the proposed methodology, and the results will be analyzed and discussed in the subsequent sections. The experimental procedures we proposed are illustrated in Fig. 1.



**Fig. 2** Proposed encoder architecture

### 3.1 Latent Vector Formation

MEDNet is a deep learning model designed to perform automated medical image analysis tasks. Models trained with an imbalanced dataset of medical images can negatively impact the accuracy and reliability of the model. To address this issue, the MEDNet model uses a technique known as latent vector formation.

Figure 2 represents the encoder component of the MEDNet architecture. It comprises a series of layers. Specifically, the encoder includes three successive Convolutional 2D layers, each of which is augmented with a Rectified Linear Unit (ReLU) activation function. Additionally, a Batch Normalization layer is incorporated after each convolutional layer to enhance the stability and convergence of the network during training. Finally, the encoder concludes with a flatten layer that outputs a latent space vector, which encodes the essential information of the input data.

The MEDNet model first inputs the medical images into a deep neural network, which extracts high-level features from the images. These features are then compressed into a lower-dimensional space, resulting in a latent vector that captures the most important information from the images. This latent vector representation provides a more compact and informative representation of the medical images, which can be used for subsequent analysis.

The latent vector formation technique has been shown to be effective in improving the accuracy of MEDNet for various medical image analysis tasks, including detecting cataracts, classifying lung nodules, and identifying breast tumors. By using this technique, the MEDNet model can better handle imbalanced datasets and achieve more accurate and reliable results.

### 3.2 Resampling Techniques

So far, we have utilized our proposed MEDNet model to obtain the latent vector from the original imbalanced dataset, as depicted in Fig. 2. In this section, our focus is directed toward addressing the training bias issue that arises due to the dataset imbalance. To mitigate this issue, we have employed several well-established resampling techniques. Initially, we utilized oversampling techniques such as Synthetic Minority Oversampling Technique (SMOTE) [15], Borderline SMOTE [16] (specifically, BorderlineSMOTE-1 for our experiment), SVM SMOTE [17], and Adaptive Synthetic Sampling (ADASYN) [18]. Subsequently, we applied undersampling techniques such as Edited Nearest Neighbor (ENN), All KNN [19], Tomek's Links [20], Neighborhood Cleaning Rule (NHC Rule) [21], and Cluster Centroid [22]. Additionally, we employed two hybrid sampling methods, namely SMOTE-ENN [23] and SMOTE Tomek [24].

### 3.3 Oversampling Methods

Datasets with unequal class distributions are a significant challenge for machine learning models. To solve this problem, many oversampling algorithms have been developed. Random oversampling, one of the earliest methods, duplicates minority class samples to achieve balance in the dataset. However, this technique can cause overfitting and a bias toward the minority class. Other oversampling techniques, such as SMOTE, ADASYN, SVM SMOTE, and Borderline SMOTE-1, have been suggested as alternatives. SMOTE generates synthetic samples by computing the k-nearest neighbors for the minority class data. ADASYN generates synthetic data for the minority class based on original samples that are misclassified by the KNN. Borderline SMOTE creates synthetic data for the minority class by considering only the borderline samples between the minority and majority classes. SVM SMOTE and Borderline SMOTE-1 are two SMOTE variants that generate synthetic data by examining the support vectors and borderline samples of optimal decision functions, respectively. These approaches can create more diverse synthetic samples and improve the model's generalization ability.

### 3.4 Undersampling Methods

To handle class imbalance in a dataset, random undersampling can be used to eliminate data samples from majority class. However, this method may lead to underfitting. To overcome this issue, controlled undersampling techniques have been proposed. The ENN method modifies the dataset by implementing the nearest neighbor algorithm and eliminates samples that don't concur with their neighbor. The All KNN

procedure, a version of ENN, considers both categories and expands the number of internal nearest neighbors in each step. The Cluster Centroid plan substitutes sample clusters by using the K-means algorithm. NHC is a fusion of ENN and KNN that focuses on cleaning data through samples discarded by ENN and the output of the three closest neighbors, irrespective of the data. This technique helps remove noise from the dataset. Tomek's Links eliminate samples that overlap between classes until they reach the minimum distance with their nearest neighbor.

### 3.5 *Hybrid Sampling Method*

Hybrid sampling is widely accepted for addressing the class imbalance problem in datasets by combining oversampling and undersampling methods. This approach can help balance the dataset by oversampling the data samples in the minority class and undersampling the majority class simultaneously, resulting in better performance. Studies have demonstrated the effectiveness of hybrid sampling over single oversampling or undersampling methods [25]. For example, the SMOTE-ENN method is a hybrid technique that employs SMOTE to oversample the minority class and after that applies ENN to eliminate noisy samples. Another hybrid method, SMOTE-Tomek, employs SMOTE to oversample the minority class and then applies Tomek's Links to remove overlapping samples between the two classes. Both of these methods demonstrated significant results in improving the classification accuracy of imbalanced datasets.

## 4 Result and Discussion

This section discusses the simulation results by conducting experiments for performance comparison of several validation classifiers on both the original imbalanced dataset and synthetic data generated by employing various resampling methods to the imbalanced latent space vector. We present a brief overview of the dataset utilized in our experiments, followed by a detailed explanation of the experimental setup. The simulations were executed on a system with a NVIDIA GeForce GTX 1650 graphics card, Ryzen 5-3550H processor, 16 GB RAM, running on Windows 10 Home 22H2, and TensorFlow 2.6.0 framework.

### 4.1 *Experimental Setup*

In our current study, we utilized a publicly available dataset [26], consisting of eight classes, namely “Normal,” “Diabetes,” “Glaucoma,” “Cataract,” “Age related Macular Degeneration,” “Hypertension,” “Pathological Myopia,” and “Other dis-

eases/abnormalities.” We have used only the “Normal” and the “Cataract” class images for our current experiment. The “Normal” class contained 1135 samples, and “Cataract” class contained 211 samples in the training set. As such, the dataset presented an imbalance problem, with “Normal” class being the majority class, and “Cataract” being the minority class.

To obtain the latent vector, the MEDNet model was trained on the dataset’s training and test samples. During unsupervised training, we normalized all input images to  $[0, 1]$  and used a combination of convolutional layers and batch normalization layer with “ReLU” activation function. Although, the latent space vectors generated by the MEDNet model had a length of 512, but they were still imbalanced. We utilized six distinct classifiers to assess the performance of the validation classifiers: decision tree (DT), support vector machine (SVM), multilayer perception (MLP), K-nearest neighbor (KNN), naïve Bayes (NB), and logistic regression (LR). We determined the model hyperparameters through grid search [27] and implemented the classifiers using the scikit-learn package [28]. For KNN classifier, the number of neighbors was set to 10, while radial basis function (RBF) was used for SVM as the kernel. We employed different configurations for each classifier. For logistic regression, “lbfgs” solver was used with the maximum iteration set to “100.” NB classifier used a variance smoothing attribute of  $1e-09$ . The MLP classifier was trained with Adam optimization algorithm using “ReLU” activation function, while keeping the learning rate at 0.001, and maximum iteration was set to 200. To judge the quality of the classifiers, we utilized various performance metrics, viz. accuracy score, precision score, recall score, and F1-score. For each experiment, we randomly partitioned the data into “Training” (80%) and “Testing” (20%). The performance of each classifier was assessed on both the original imbalanced data and modified data generated by applying different sampling methods.

## 4.2 Comparison of Classifiers

Within this section, we will assess the effectiveness of the suggested framework by utilizing evaluation metrics, including precision, recall, accuracy, and F1-score. These metrics are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1 Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

In the given equation, TP refers to the number of true positive predictions, TN refers to the number of true negative predictions, FP refers to the number of false positive predictions, and FN refers to the number of false negative predictions.

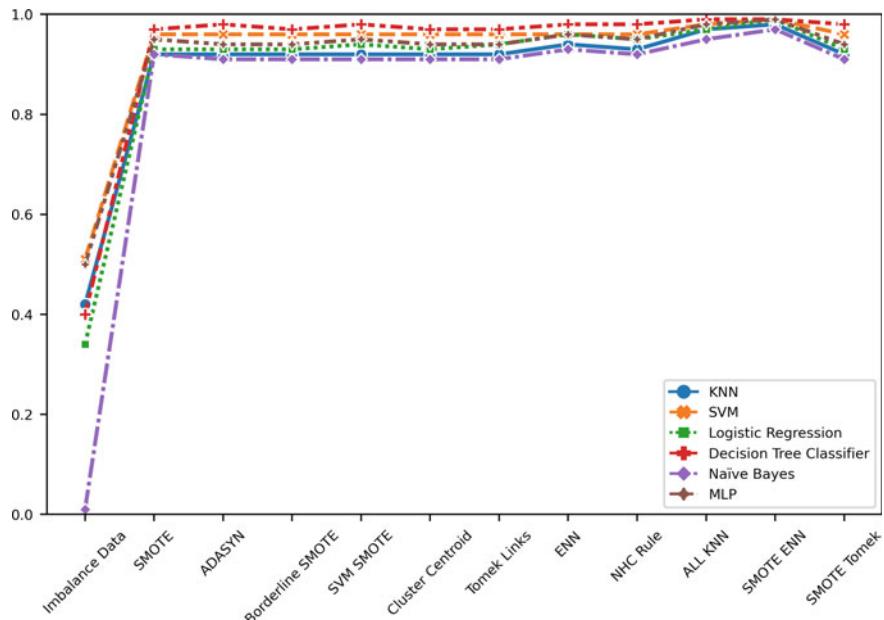
The performance evaluation of a MEDNet-based cataract detection system, which employs various sampling strategies for training, is presented in this study. The accuracy scores of different classification algorithms for the original imbalance data and the synthetic data are shown in Fig. 3.

The results obtained from the experiments show that the original imbalanced dataset produced unsatisfactory accuracy scores for the LR, SVM, DT, NB, KNN, and MLP classifiers. However, the application of the SMOTE oversampling method resulted in a noteworthy improvement in the performance of the classifiers. Satisfactory results were obtained from various other sampling techniques such as ADASYN, Cluster Centroid, Tomek Links, NHC Rule, Borderline SMOTE, SVM SMOTE, All KNN, ENN, and SMOTE Tomek. As illustrated in Fig. 3, the SMOTE-ENN hybrid sampling method achieved the highest performance score across all classifiers.

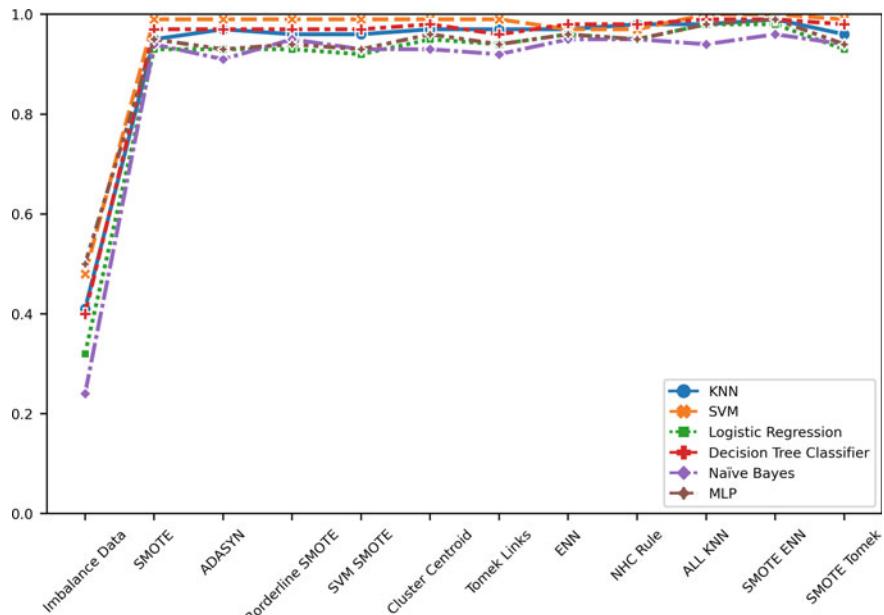
The precision, recall, and F1-scores also demonstrated a marked improvement after applying the SMOTE oversampling method, as illustrated in Figs. 4, 5, and 6. The overall recall, precision, and F1-scores for all classifiers increased by 60, 41, and 74%, respectively. These results show that the resampling methods were effective in addressing the imbalanced nature of the dataset and resulted in significant performance improvement compared to the original imbalance data.

### **4.3 Comparative Analysis with State of the Art**

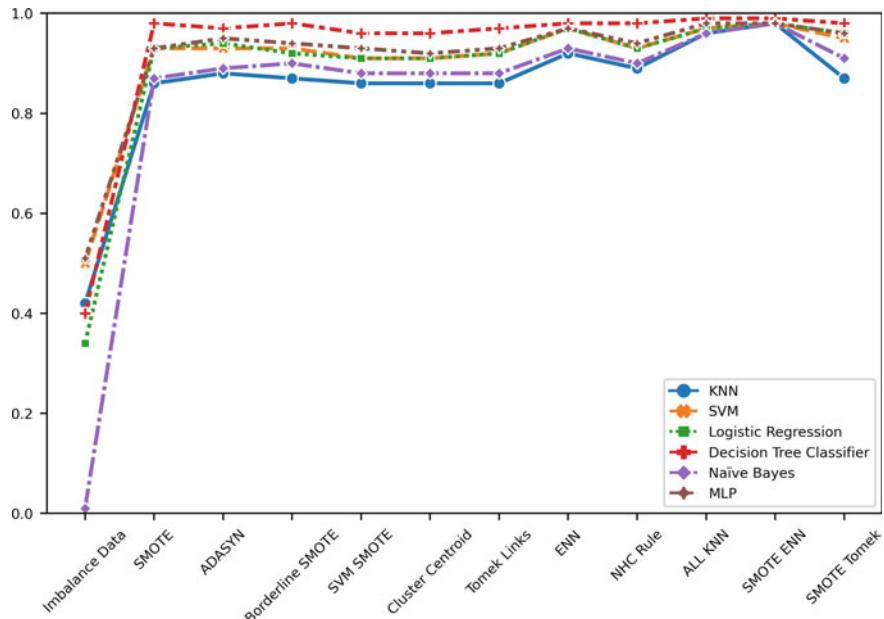
To assess the efficacy of our proposed approach for detecting imbalanced cataracts using ophthalmic images, we conducted a comparative analysis with several previous studies that employed different machine learning algorithms and architectures. The findings are presented in Table 1 that summarizes the performance of our approach and that of earlier studies, based on various evaluation metrics such as accuracy, precision, recall, and F1-score. The bold font in Table 1 denotes the most optimal outcomes. Our approach exhibited superior performance in terms of all performance metrics, demonstrating its effectiveness for imbalanced cataract detection using ophthalmic images compared to earlier studies.



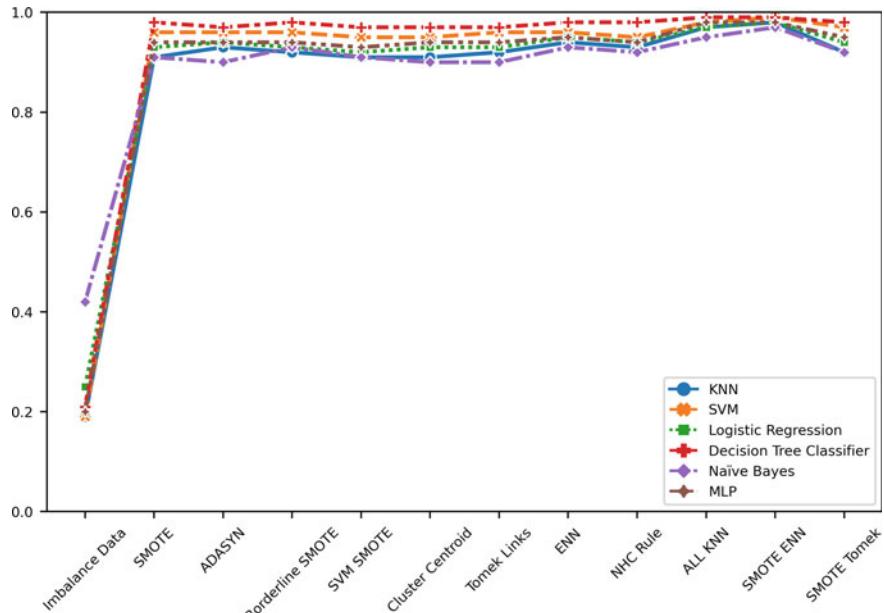
**Fig. 3** Performance analysis based on accuracy score



**Fig. 4** Performance analysis based on precision score



**Fig. 5** Performance analysis based on recall score



**Fig. 6** Performance analysis based on F1-score

**Table 1** Comparison with existing state-of-the-art techniques

Study	Accuracy	Precision	Recall	F1-score
Jiang et al. [13]	0.92	0.89	0.93	0.86
Imran et al. [29]	0.96	0.95	0.99	0.96
Elloumi et al. [30]	0.93	0.91	0.95	0.94
Proposed method	<b>0.99</b>	<b>1</b>	<b>0.99</b>	<b>0.99</b>

## 5 Conclusion

In this study, we focused on detecting cataracts from ophthalmic images using MED-Net. Our approach involved training the MEDNet model on an imbalanced dataset of ophthalmic images to obtain a feature representation of the images in the form of a latent vector, which also suffered from biasing issues due to the imbalanced data. To overcome this, we utilized various resampling techniques, including oversampling, undersampling, and hybrid sampling algorithms, to create a balanced dataset. After balancing the dataset, we proceeded to train and assess the performance of various classifiers. Our experimental results showed that using the proposed approach of combining MEDNet with resampling techniques significantly improved the performance of the classifiers. Our study revealed that the SMOTE-ENN hybrid sampling technique demonstrated superior classification performance compared to other sampling methods, indicating that it can be a powerful tool for the accurate diagnosis and treatment of this vision problem. Automated cataract detection methods that address the class imbalance problem and employ advanced deep learning techniques can improve early diagnosis and treatment, leading to better vision outcomes for patients.

## References

1. U.S. Department of Health and National Eye Institute Human Services (2022) Eye health data and statistics. <https://www.nei.nih.gov/learn-about-eye-health/eye-health-data-and-statistics>. Last Accessed on 10 Feb 2023
2. Steinmetz JD, Bourne RRA, Briant PS, Flaxman SR, Taylor HRB, Jonas JB, Abdoli AA, Abraha WA, Abualhasan A, Abu-Gharbieh EG et al (2021) Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: an analysis for the global burden of disease study. Lancet Glob Health 9(2):e144–e160
3. Guo Y, Camino A, Wang J, Huang D, Hwang TS, Jia Y (2018) Mednet, a neural network for automated detection of avascular area in oct angiography. Biomed Opt Exp 9(11):5147–5158
4. Zhang X-Q, Hu Y, Xiao Z-J, Fang J-S, Risa H, Jiang L (2022) Machine learning for cataract classification/grading on ophthalmic imaging modalities: a survey. Mach Intell Res 19(3):184–208

5. Goh JHL, Lim ZW, Fang X, Anees A, Nusinovici S, Rim TH, Cheng C-Y, Tham Y-C (2020) Artificial intelligence for cataract detection and management. *Asia-Pac J Ophthalmol* 9(2):88–95
6. Jindal I, Gupta P, Goyal A (2019) Cataract detection using digital image processing. In: 2019 global conference for advancement in technology (GCAT). IEEE, pp 1–4
7. Junayed MS, Islam MdB, Sadeghzadeh A, Rahman S (2021) Cataractnet: an automated cataract detection system using deep learning for fundus images. *IEEE Access* 9:128799–128808
8. Shehzad M, Qadri S, Aslam T, Qadri SF, Razzaq A, Muhammad SS, Ali Nawaz S, Ahmad N (2020) Machine vision based identification of eye cataract stages using texture features. *Life Sci J* 17(8):44–50
9. Mitra A, Roy S, Roy S, Setua SK (2018) Enhancement and restoration of non-uniform illuminated fundus image of retina obtained through thin layer of cataract. *Comput Methods Programs Biomed* 156:169–178
10. Luo Y, Kun C, Lei L, Liu J, Mao J, Ke G, Sun M (2020) Dehaze of cataractous retinal images using an unpaired generative adversarial network. *IEEE J Biomed Health Inform* 24(12):3374–3383
11. Jiang J, Wang L, Fu H, Long E, Sun Y, Li R, Li Z, Zhu M, Liu Z, Chen J et al (2021) Automatic classification of heterogeneous slit-illumination images using an ensemble of cost-sensitive convolutional neural networks. *Ann Transl Med* 9(7)
12. Bhati A, Gour N, Khanna P, Ojha A (2023) Discriminative kernel convolution network for multi-label ophthalmic disease detection on imbalanced fundus image dataset. *Comput Biol Med* 106519
13. Jiang J, Liu X, Zhang K, Long E, Wang L, Li W, Liu L, Wang S, Zhu M, Cui J et al (2017) Automatic diagnosis of imbalanced ophthalmic images using a cost-sensitive deep convolutional neural network. *Biomed Eng Online* 16(1):1–20
14. Jun TJ, Eom Y, Kim C, Kim D et al (2019) Tournament based ranking cnn for the cataract grading. In: 2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE, pp 1630–1636
15. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
16. Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Advances in intelligent computing: international conference on intelligent computing, ICIC 2005, Hefei, China, 23–26 Aug 2005, Proceedings, Part I. Springer, pp 878–887
17. Nguyen HM, Cooper EW, Kamei K (2011) Borderline over-sampling for imbalanced data classification. *Int J Knowl Eng Soft Data Paradigms* 3(1):4–21
18. He H, Bai Y, Garcia EA, Li S (2008) Adasyn: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, pp 1322–1328
19. Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans Syst Man Cybern* 3:408–421
20. Elhassan T, Aljurf M (2016) Classification of imbalance data using Tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. *Glob J Technol Optim S* 1:2016
21. Laurikkala J (2001) Improving identification of difficult small classes by balancing class distribution. In: Artificial intelligence in medicine: 8th conference on artificial intelligence in medicine in Europe, AIME 2001 Cascais, Portugal, July 1–4, 2001, Proceedings 8. Springer, pp 63–66
22. Radev DR, Jing H, Styś M, Tam D (2004) Centroid-based summarization of multiple documents. *Inform Process Manage* 40(6):919–938
23. Batista GE, Bazzan ALC, Monard MC et al (2003) Balancing training data for automated annotation of keywords: a case study. *WOB* 10–18
24. Batista GE, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor News* 6(1):20–29

25. Chatterjee S, Maity S, Bhattacharjee M, Banerjee S, Das AK, Ding W (2022) Variational autoencoder based imbalanced covid-19 detection using chest x-ray images. New Gener Comput 1–36
26. Larxel (2020) Ocular disease recognition. <https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k>. Last Accessed on 10 Feb 2023
27. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(2)
28. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. J Mach ine Learning research 12:2825–2830
29. Imran A, Li J, Pei Y, Mokbal FM, Yang J-J, Wang Q (2020) Enhanced intelligence using collective data augmentation for cnn based cataract detection. In: Frontier computing: theory, technologies and applications (FC 2019). Springer, vol 8, pp 148–160
30. Yaroub E (2022) Cataract grading method based on deep convolutional neural networks and stacking ensemble learning. Int J Imag Syst Technol 32(3):798–814

# A Supervised Learning Algorithm for Disease Prediction Using Complex Network and Machine Learning



Sangita Dutta, Navneet Singh, Susmita Das, and Susanta Chakraborty

**Abstract** The availability of advanced tools to store and process huge volumes of data has propelled the healthcare domain to adopt a more data-driven approach to disease detection and prevention. Data mining and machine learning are major contributors to designing prediction models that can accurately predict diseases but they overlook the connection between diseases and symptoms. These relationships can be represented and understood using network analysis. In this study, we have proposed a novel complex network-based approach that classifies a patient based on their medical test data and symptoms. Each node in the network represents a patient, and a similarity score is calculated for each patient pair in the network using the vector representation of their diagnosis data. Connections between the nodes are made using this similarity score such that patients with similar diagnoses and symptoms are connected to each other. Using this Patient Similarity Network an instance-based learning approach to classify new instances has been built. Every node in the network has a weighted contribution toward classifying the new instance. For evaluating the proposed method, the prediction accuracy is calculated which is found to be good. We perform a comparative study with existing prediction algorithms like KNN and SVM for validation purposes.

**Keywords** Disease prediction · Arrhythmia · Patient network · Complex network · Degree centrality

---

S. Dutta (✉) · S. Das · S. Chakraborty

Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India  
e-mail: [duttasangita72@gmail.com](mailto:duttasangita72@gmail.com)

S. Das

e-mail: [susmitad900@gmail.com](mailto:susmitad900@gmail.com)

S. Chakraborty

e-mail: [sc@cs.iiest.ac.in](mailto:sc@cs.iiest.ac.in)

N. Singh

Moneyview, Bangalore, Karnataka 560087, India  
e-mail: [navneesingh3995@gmail.com](mailto:navneesingh3995@gmail.com)

## 1 Introduction

Significant advancements in the field of health and medicine have made the cure of many incurable diseases possible. Substantial research has been carried out to discover strategies that make the process of early detection easier. With the quantity of digital healthcare data collected daily, various prediction techniques such as machine learning and data mining are extensively used to detect health hazards at an early stage to ensure that the patient receives proper care. The use of network analysis has gained interest, allowing researchers to model the relationship between data points and providing information crucial for accurate prediction results. Chronic diseases do not always occur in isolation and thus have various common risk factors that can be demographic, environmental, or ancestral. In such cases, a network-based method is more suited to the realm of disease risk prediction since it takes into consideration the link between diseases and their symptoms. In this study, a generic supervised learning model using complex network [2, 3] theory has been implemented for chronic disease prediction. The objectives of this study are:

1. We develop a methodology to represent patient data as a network.
2. Generated a prediction model that can determine if a new patient is healthy or at risk of developing a chronic condition.

Among various chronic diseases, cardiovascular diseases have long been a source of concern for people of all ages because of their unhealthy lifestyle habits like lack of exercise, smoking, substance use, and alcohol. Cardiac arrhythmia is one such condition, also known as cardiac rhythm issue, wherein the patients experience irregularities in their heart rate. Although not all forms of arrhythmias are dangerous, they can occasionally signify a serious heart condition. Physicians utilize a 12-lead electrocardiogram (ECG) to diagnose and monitor the kind of arrhythmia a patient has because the symptoms can be relatively subtle in most situations. The range and interval of several ECG parameters, such as heart rate, PR waves, and QRS complex, are evaluated to discover the underlying cause of abnormal cardiac function.

In this paper, we used our proposed framework, inspired by complex network theory, to predict the type of arrhythmia a patient might have from the arrhythmia patients dataset. The dataset provides ECG readings of the patients for diagnosing the type of arrhythmia they might have. The performance of our proposed method has been validated by comparing it with existing prediction methods such as *k-nearest neighbors* (KNN) [8], *support vector machine* (SVM) [17], *decision trees* (DT) [16], and *Naive Bayes* (NB) [20]. Comparative research utilizing several other patient datasets available in the repository has been performed. Extending this study, we devised a probabilistic estimate of the risk of heart failure among already diagnosed arrhythmia patients.

## 2 Literature Review

With an aim to improve healthcare facilities and provide better and timely diagnosis, disease risk prediction is recently the most explored in the area of bioinformatics. Many statistical and data mining approaches have been suggested by researchers using historical medical data, but these do not explicitly take into account the relationship between diseases and their symptoms.

Guvenir et al. [10] in their study present a new supervised feature projection-based algorithm that classifies based on feature votes. It outperforms other standard algorithms such as Naive Bayesian and nearest neighbors classifier. Chen et al. [4] proposed a new *CNN-based* disease prediction using structured and unstructured hospital data. The goal is to predict whether a patient is in high-risk population of cerebral infection. Steinhaeuser et al. [18] present a proactive healthcare approach in their study, and the proposed methodology uses diagnosis data of around 13 million patients to construct a network of comorbid diseases. [6] studied diagnostic medical data and converted this data into a phenotypic network.  $\phi - correlation$  is used to quantify the strength of the relationship between two diseases. Gao et al. [7] established a disease similarity scoring function which is an improved version of the available conventional methods. In order to construct a single homogeneous disease network, the intersection of nodes from all these disease networks is taken. Dursan et al. [5] presented a converted study of prediction breast cancer survivability using machine learning as well data mining techniques. Ahmad et al. [1] use multiple machine learning algorithms like GBC, SVM, KNN, and LR along with Grid-SearchCV for the prediction of cardiac disease. A fivefold cross-validation technique is used by the system for verification. Gupta et al. [9] apply principal component analysis (PCA) on medical data for the extraction of significant features. Random forest classifier has been implemented to predict Parkinson's disease. Mahesh et al. [14] use relevant features for the construction of an ensemble learning system based on Bayesian networks for predicting and treating chronic diabetes. To capture the latent graph structure, Zheng et al. [21] used an adaptive graph learning approach. A multi-modal graph learning model (MMGL) for disease prediction with multi-modality was proposed and applicable to inductive learning environments. Adaptive graph learning (AGL) is suggested to obtain a latent network structure to match flexibly for GNN-based downstream tasks in order to discover the inherent relationships among patients. Kalgotra et al. [12] propose a model that can assist in predicting the patients with a risk of developing colorectal cancer and accordingly provide diagnostic steps for these patients. The model employs the suggested variables generated through comorbidity network matrices acquired from colorectal cancer and non-colorectal cancer patients.

### 3 Dataset

The dataset used for this study is taken from the University of California (UCI) repository for machine learning. It contains diagnostic data of 452 patients with 279 ECG parameters obtained from their 12-Lead ECG examination. Based on the diagnosis done by an expert cardiologist, each patient is categorized into 16 different classes. Class 1 is for patients not exhibiting any symptoms of arrhythmia falling under the *Normal* category, whereas Classes 2–15 are different types of arrhythmias. All the other patients belong to Class 16. Details of classes and their distribution in the dataset are given in Table 1.

The ECG machine measures the electrical activity of the heart and generates a graph of the voltage over time. For a 12-lead ECG, 10 different sensors or electrodes are placed on the patient's body. *Lead I* measure the voltage between the right arm and the left arm, *Lead II*: right arm and left leg, *Lead III*: left arm and left leg. *V1–V6* is called the chest leads and is positioned around the patient's rib cage. The remaining three leads are augmented limb leads *aVR* (Augmented Vector Right), *aVL* (Augmented Vector Left), *aVF* (Augmented Vector Foot) measured over the right shoulder, left shoulder, and the foot, respectively. Further, the ECG curve is divided into five segments P, Q, R, S, and T. The dataset provides the intervals, amplitude, and wave widths of all these segments for all the 12 leads. Some physical features like age, gender of the patient, height, and weight are also present. A detailed description of all the features is given in Table 2. Issues with the raw dataset were resolved using various data-cleaning techniques which are discussed in detail in the upcoming sections.

**Table 1** Class distribution of arrhythmia dataset

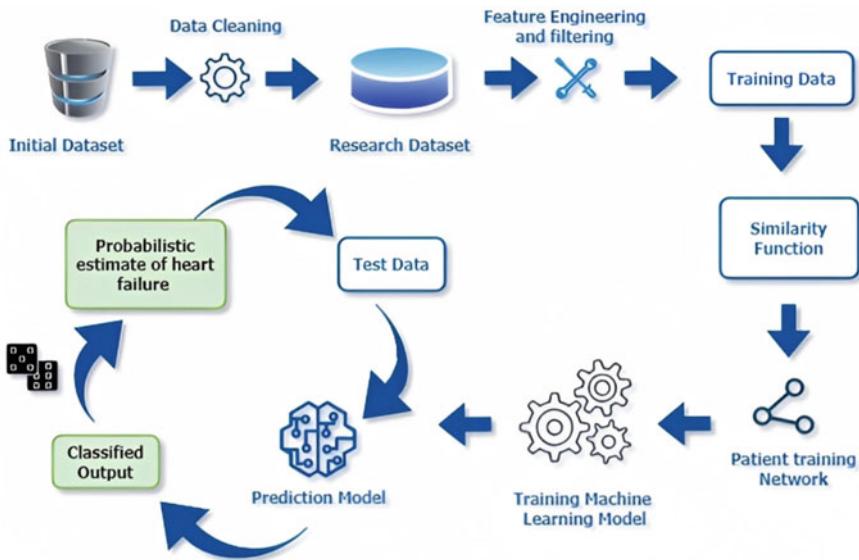
Class ID	Class label	Frequency
01	Normal	56.4
02	Coronary artery disease	8.6
03	Old anterior myocardial infarction	3.1
04	Old inferior myocardial infarction	3.3
05	Sinus tachycardia	3.1
06	Sinus bradycardia	5.7
07	Ventricular premature contraction	0.5
08	Supra-ventricular premature contraction	0.5
09	Left bundle branch block	2.1
10	Right bundle branch block	11.4
11	1st degree AV block	0
12	2nd degree AV block	0
13	3rd degree AV block	0
14	Left vertical hypertrophy	0.9
15	Atrial fibrillation	0
16	Others	4.2

**Table 2** Details of attributes in the arrhythmia dataset

S. No.	Attribute name	Details
F1	Age	Age of the patient
F2	Sex	Gender of the patient: 0—Male, 1—Female
F3	Height	Height of patient in cms
F4	Weight	Weight of the patient in kgs
F5	QRS duration	Time for ventricular depolarization
F6	P–R interval	The period of time from the onset of the P wave to the beginning of the QRS complex is termed the P–R interval
F7	Q–T interval	Linear the Q–T interval is the time from the beginning of the QRS complex, representing ventricular depolarization, to the end of the T wave
F8	T interval	The T wave represents ventricular repolarization
F9	P interval	P interval represents the electrical depolarization of the atria of the heart
F10	QRS	The QRS complex represents the electrical impulse as it spreads through the ventricles and indicates ventricular depolarization
F11	p	The P wave is the first positive deflection on the ECG curve
F12	T	T wave is the positive deflection after each QRS complex
F13	QRST	
F15	J	The J point is a point in time marking the end of the QRS and the onset of the ST segment present on all ECG's
F16	Heart rate	Heart beats per minute of the patient
F17–F20	Average wave width	Average width of Q, R, S, R' and S' waves
F21	Intrinsic deflections	The R wave peak time, is measured from the beginning of the QRS complex to the peak of the R wave
F22–F27	Existence of ragged/diphasic deflection of waves	Existence of Ragged waves or diphasic deflections in the R, P, and T waveforms
F28–F35	Wave amplitude	Wave amplitude for JJ, Q, R, S, R', S', P, and T waves
F36	QRSA	Sum of areas under the curve of all segments
F37	QRSTA	Calculated as $(\text{QRSA} + 0.5 * \text{Width}_{\{\text{T wave}\}} * 0.1 * \text{Height}_{\{\text{T wave}\}})$

#### 4 Proposed Method

A two-step approach has been established, to achieve optimal prediction results of whether our subject has symptoms of cardiac arrhythmia.



**Fig. 1** Overall framework of predicting risk of heart failure

#### 4.1 Overview

A patient-similarity network is constructed by establishing a relationship between the patient data. Using this similarity network as input to our proposed prediction model the presence or absence of cardiac arrhythmia in our subject is predicted. The overall framework has been depicted in Fig. 1. A complex network is constructed from raw data by establishing a relationship between the patients. Targeting the idea that patients diagnosed with similar diseases exhibit similar symptoms, a patient-similarity network  $N_p$  is constructed. For each patient pair, a similarity score is calculated and the pairs with higher scores are connected in the network. Use the constructed network  $N_p$  to predict if the test patient has arrhythmia or not. The underlying methodology is inspired by instance-based learning algorithms where a limited set of attributes are used to find neighboring nodes (patients) in the network that are similar (with respect to their symptoms) to the new patient. All such nodes then vote toward classifying this patient. A detailed description of the method is discussed in the following sections.

#### 4.2 Patient Network

A patient network  $N_p$  presents the interaction among the patients based on the similarities or associations between their symptoms and medical diagnosis. In our dataset,

each patient (instance) forms a new node in the network. Edges are constructed after studying the associations and similarities among the nodes. Formally, the network  $N_p$  be represented as follows:

- The set  $X = \{x_1, x_2 \dots x_n\}$  where  $n$  is the number of instances in the training data has a one-to-one correspondence with the set of vertices in the PSN represented as  $V = \{v_1, v_2 \dots v_n\}$ .
- Each vertex  $v_i$  in  $V$  is represented by a set of features  $F_i = \{f_1, f_2 \dots f_m\}$  where  $m$  is the number of features in the dataset, and a label  $c_i$  belonging to the set of labels  $C_i = \{c_1, c_2 \dots c_m\}$  where  $m$  is the number of distinct labels the dataset is divided into.
- A similarity matrix  $S$  is constructed such that the element  $s_{ij}$  of the matrix gives the similarity score [13] of vertices  $v_i$  and  $v_j$  belonging to the set  $V$  using their vector-representations  $V_i$  and  $V_j$ , respectively, shown in Eq. 1.

$$s_{ij} = \begin{cases} f_{\text{sim}}(V_i, V_j), & \text{if } f_{\text{sim}}(V_i, V_j) \geq s_{\text{threshold}} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where

$$f_{\text{sim}} = \frac{V_i * V_j}{|V_i| + |V_j|} \quad (2)$$

The notion of employing cosine similarity to calculate the similarity score allows for catching the semantic as well as the directional similarities among the patient vectors. Similarity score calculation is shown in Eq. 2.

- The edge set  $E$  of the network  $N_p$  is constructed using the above calculated similarity matrix  $S$ . If  $s_{ij} \neq 0$ , edge  $e_{ij}$  is added to the edge set  $E$ .
- The patient network allows for establishing a relationship between their symptoms which increases the probability of them being diagnosed with the same disease. New patient added to this network will lie in close proximity to the ones diagnosed with a particular disease, making it highly probable for the new patient to acquire the same.

### 4.3 Disease Prediction Model

This part of the framework essentially classifies a new patient based on the similarity of symptoms and medical parameters with the patient network  $N_p$ . It is implemented in three sequential phases discussed below.

- Insert a new patient node into the patient network and find its k-similar neighboring nodes. This is analogous to the traditional k-nearest neighbor approach.
- A contribution factor (CF) is calculated for each of the nodes in the neighbor set using the degree centrality [19] of the node. Nodes with higher degree centrality

are the nodes that exhibit similar symptoms to the majority of the patient nodes in the neighborhood, contributing more toward predicting the type of arrhythmia our test patient has. A contribution factor (CF) is calculated using Eq. 3.

$$\text{CF}_i = \frac{d(v_i)}{|V| - 1} \quad (3)$$

where  $d(v_i)$  is the degree of the vertex  $i$  in the network  $N_p$ .

- The contribution factor (CF) to the corresponding labels is added by traversing all the nodes in the neighbor set. The maximum weight label is then applied to the new patient node.

## 5 Experimental Results

This section discusses the performance of the proposed prediction algorithm using the dataset of cardiac arrhythmia patients from the University of California, Irvine (UCI) machine learning repository. For a comprehensive study of the accuracy of the model, the model is implemented on other datasets as well.

### 5.1 Results on Arrhythmia Dataset

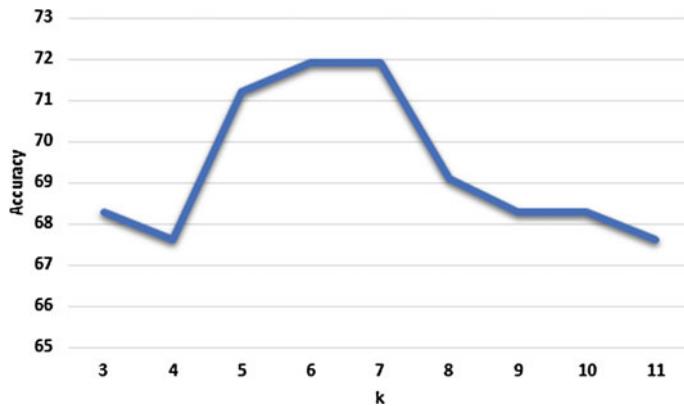
For evaluation, the prediction accuracy is calculated shown in Eq. 4. It is the most commonly used metric and gives the percentage of test instances that have been correctly classified by the model. So, higher accuracies are expected from the model to be suitable for prediction. It can be easily calculated using the confusion matrix.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (4)$$

The proposed model achieved a maximum prediction accuracy of 71.9% for  $k = 6$  where  $k$  is the number of neighbors. To identify the most optimal value of  $k$ , it is observed that there is a change in accuracy with increasing values of  $k$  starting from  $k = 3$  to  $k = 11$ . The values are summarized in Table 3, and the resultant curve is given in Fig. 2. It can be safely concluded that with increasing values of  $k$ , the accuracy improves up to a threshold value (in our case  $k_{\text{threshold}} = 7$ ) and after that starts deteriorating.

**Table 3** Accuracy measure of the proposed method with varying values of  $k$  and  $l = 1$

$K$	3	4	5	6	7	8	9	10	11
Accuracy	68.3	67.6	71.2	71.9	71.9	69.1	68.3	68.3	67.6



**Fig. 2** Accuracy plot relative to the increasing  $k$  values

**Table 4** Accuracy comparison of existing classifiers and our proposed method using multiple datasets

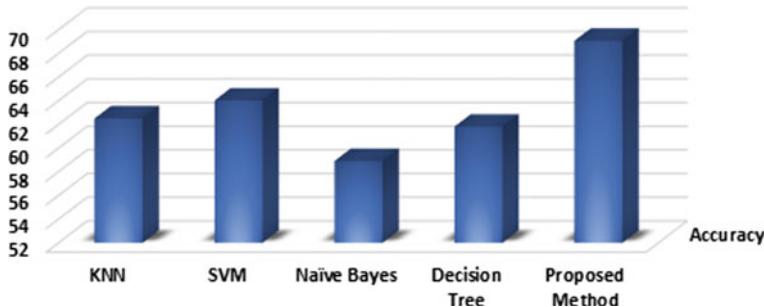
Dataset/model	KNN	SVM	Decision tree	Naïve Bayes	Proposed method
Arrhythmia	62.59	64.02	58.99	61.87	<b>69.06</b>
<i>E. Coli</i> [15]	80.18	74.77	78.37	69.33	<b>81.08</b>
Thyroid [11]	95.78	87.32	94.36	92.95	<b>97.18</b>

Note: Our Proposed method gives the best results than other existing methods

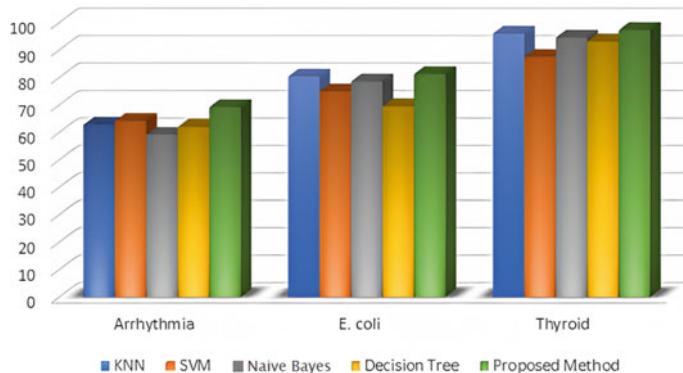
We also did a comparison study with other prevalent models like KNN, SVM, and Naïve Bayes. Table 4 shows the comparison of the accuracy of all these models when executed on the arrhythmia dataset. Keeping  $k = 3$  for KNN as well as the proposed method, it is evident that the proposed method has the maximum prediction accuracy of 69.06%, whereas naïve Bayes shows the worst accuracy of 58.99% in predicting the presence of cardiac arrhythmia in a patient. Although the model gives accurate results for smaller and medium-sized datasets, larger datasets might reduce the efficiency of the training model. With huge amounts of patient data, constructing a patient-similarity network (PSNs) might take up huge computational power.

## 5.2 Results from Other Datasets

To comprehensively study the performance of our model, it is executed on different disease datasets. These datasets are open source and available in the University of California, Irvine (UCI) machine learning repository. The resultant accuracies are compiled in Table 4. As evident, the proposed method outperforms all other classifiers for all datasets that were considered for this comparison.



**Fig. 3** Accuracy comparison of existing classifiers with our proposed method



**Fig. 4** Prediction accuracy comparison of multiple prediction models using arrhythmia, *E. coli*, and thyroid datasets

To validate the impact of degree centrality on the classification, we compare the accuracies from two different scenarios of our algorithm. In the first scenario, the class weights are calculated using equal votes from each of the neighboring nodes. In the second scenario, the importance of each node is enumerated using its degree of centrality. A graph of the accuracies is plotted for the number of neighbors varying from 3 to 13. In Figs. 3 and 4, the accuracies obtained are plotted from both these variations. As it is evident that although the curves obtained are the same, there is a significant difference in the performance of the model.

## 6 Conclusion and Future Work

In this paper, we study the results from the ECG examination of patients to design an appropriate algorithm for predicting arrhythmia using complex network metrics along with the concept of instance-based learning. A patient-similarity network is

constructed from the raw dataset based on the similarity between their symptoms. Using this network, a  $k$ -similar-node set is put together wherein each node contributes toward classifying a new patient. The results of this study confirm that the use of complex networks ensures better disease prediction results than the pre-existing machine learning models. For the dataset under consideration, our model gives a maximum accuracy of 71.9%. We can safely conclude that the major advantage of using the proposed approach is that it gives a holistic view of the relationship between the patients and their diagnosis. Quantifying this relationship has given a major boost to the accuracy of the model. Using patient networks for disease prediction is comparatively new paradigm. Another major concern is that as the patient data size increases, the computational power also increases, thereby reducing the efficiency. This provides a lot of scope for future work for more scalable and efficient solution that can be applied across multiple datasets. The future scope of this study would be to test the applicability of this model over time-variant disease datasets.

## References

1. Ahmad GN, Fatima H, Ullah S, Saidi AS et al (2022) Efficient medical diagnosis of human heart diseases using machine learning techniques with and without gridsearchcv. IEEE Access 10:80151–80173
2. Barabási RAAL (2002) Statistical mechanics of complex networks. Rev Modern Phys 74(1):47–97
3. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU (2006) Complex networks: structure and dynamics. Phys Rep 424(4–5):175–308
4. Chen M, Hao Y, Hwang K, Wang L, Wang L (2017) Disease prediction by machine learning over big data from healthcare communities. IEEE Access 5:8869–8879
5. Delen D, Walker G, Kadam A (2005) Predicting breast cancer survivability: a comparison of three data mining methods. Artif Intell Med 34(2):113–127
6. Folino F, Pizzuti C, Ventura M (2010) A comorbidity network approach to predict disease risk. Information technology in bio-and medical informatics, ITBAM 2010: first international conference, Bilbao, Spain, 1–2 Sept 2010. Springer, Berlin, pp 102–109
7. Gao J, Tian L, Wang J, Chen Y (2020) Song Bo, Hus X (2020) Networks SDPWHDI. IEEE Transactions on Nanobioscience 19(3):571–578
8. Guo G, Wang H, Bell D, Bi Y, Greer K (2003) Knn model-based approach in classification. In: On the move to meaningful internet systems 2003: CoopIS, DOA, and ODBASE: OTM confederated international conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy. Proceedings, 3–7 Nov 2003. Springer, pp 986–996
9. Gupta I, Sharma V, Kaur S, Singh AK (2022) PCA-RF: an efficient parkinson's disease prediction model based on random forest classification. arXiv preprint [arXiv:2203.11287](https://arxiv.org/abs/2203.11287)
10. Guvenir HA, Acar B, Demiroz G, Cekin A (1997) A supervised machine learning algorithm for arrhythmia analysis. In: Computers in cardiology. IEEE, pp 433–436
11. Institute G (1987) Thyroid disease data set. <https://archive.ics.uci.edu/ml/datasets/thyroid+disease>
12. Kalgotra P, Sharda R, Parasa S (2023) Quantifying disease-interactions through co-occurrence matrices to predict early onset colorectal cancer. Decis Support Syst 113929
13. Lahitani AR, Permanasari AE, Setiawan NA (2016) Cosine similarity to determine similarity measure: study case in online essay assessment. In: 2016 4th international conference on cyber and IT service management. IEEE, pp 1–6

14. Mahesh T, Kumar D, Vinoth Kumar V, Asghar J, Mekcha Bazezew B, Natarajan R, Vivek V (2022) Blended ensemble learning prediction model for strengthening diagnosis and treatment of chronic diabetes disease. *Comput Intell Neurosci*
15. Nakai K (1996) Ecoli data set. <https://archive.ics.uci.edu/ml/datasets/ecoli>
16. Safavian SR, Landgrebe D (1991) A survey of decision tree classifier methodology. *IEEE Trans Syst Man Cybern* 21(3):660–674
17. Schuld C, Laptev I, Caputo B (2004) Recognizing human actions: a local svm approach. In: Proceedings of the 17th international conference on pattern recognition. ICPR 2004. IEEE, vol 3, pp 32–36
18. Steinhaeuser K, Chawla NV (2009) A network-based approach to understanding and predicting diseases. *Social computing and behavioral modeling*. Springer, Berlin, pp 1–8
19. Vairachilai S, Kavitha Devi MK, Raja M (2017) Analysis of statistical and structural properties of complex networks with random networks. *Appl Math Inform Sci* 11(1):137–146
20. Yang FJ (2018) An implementation of naive bayes classifier. In: 2018 international conference on computational science and computational intelligence (CSCI). IEEE, pp 301–306
21. Zheng S, Zhu Z, Liu Z, Guo Z, Liu Y, Yang Y, Zhao Y (2022) Multi-modal graph learning for disease prediction. *IEEE Trans Med Imag* 41(9):2207–2216

# Deep Learning-Based Real-Time Hand Gesture Recognition Using Histogram of Oriented Gradient



Anurag Sahu, Tannistha Pal, and Suman Deb

**Abstract** The main social issues facing the deaf-dumb population that keep them from using the most basic and necessary services of life are the communication gap and the hearing majority. Even though the issue has been addressed by advancements in automatic sign language identification, an effective solution has still not been found as a result of a number of difficult issues. The majority of works currently in existence attempt to create vision-based recognition system using the traditional pattern analysis approach by deriving intricately designed feature descriptors from the gesture's images. However, the effectiveness of those techniques is severely constrained when dealing with a substantial sign vocabulary recorded in chaotic and complicated backdrop settings. This paper introduces an algorithm based on histogram of oriented gradient (HOG) and support vector machine (SVM) for detecting the hand gestures. Further, we have also used convolution neural network to classify different types of hand gestures. Visual hand gesture recognition is becoming more and more desired for user interfaces for human–computer interaction. The experimental outcomes demonstrate that the proposed approach shows its effectiveness in terms of an accuracy of 80%, precision of 0.84, recall of 0.81, and F1-score of 0.83 compared with other existing hand gesture recognition techniques. Hence, this proposed technique could be employed as a viable route for hand gesture detection, while lowering computational cost and time.

**Keywords** Hand gesture · SVM · HOG · Motion capture · Evaluation metrics

---

A. Sahu

Department of Biotechnology and Bioinformatics, Vignan Foundation For Science Technology and Research, Vadlamudi, Andhra Pradesh 522213, India

T. Pal () · S. Deb

Department of Computer Science and Engineering, National Institute of Technology, Agartala, Barjala, Jirania 799046, India

e-mail: [tannisthapaul@gmail.com](mailto:tannisthapaul@gmail.com)

## 1 Introduction

A significant component of human–computer interaction is hand gesture recognition. Technological advancements are developing quickly right now, making it easy for organizations to incorporate new concepts and innovations to increase output and efficiency. Although any physical motion or state might result in a gesture, the hand or the face is the most common places for them to begin. The idea behind gesture recognition is that it gives computers a means to interpret human expression. As a result, there is less need for text-based and graphical user interfaces. In order for a gesture to be understood by another person, it needs to be conveyed in some form of message. Typically, a gesture is considered of as a motion of a body part, often a hand or the head, to convey a thought or message [1, 2].

Recent years have seen significant progress in the development of user-computer interfaces that are intelligent, smart, and on various human gestures recognition [3–6, 11–14]. Gestures signify a straightforward confluence for both users and machines. As a result, gesture-based interfaces can be utilized to both replace and enhance existing standard interface devices. Modern gesture recognition technologies are much more recent. In recent years, the technology has started to spread across a number of industries because of advancements in computer vision, sensors, machine learning, and deep learning that have boosted the technology's accessibility and accuracy.

The organization of the paper is depicted as follows. Sect. 2 describes the literature survey, in which the existing works regarding hand gesture recognition are considered. Later, Sect. 3 describes the framework proposed, and in Sect. 4, the results and analysis of the experiments performed are discussed, followed by Sect. 5 where the research work is concluded in.

## 2 Literature Review

Visual hand tracking and recognition are a difficult problem to solve. There has been intensive research into hand gesture recognition during the past three decades. The majority of the effort has used either a non-vision approach or a vision-based method. The non-vision-based strategy captures data via interfaces including data gloves, motion sensors, and position trackers. Although this method works, the hardware setup is costly and inconvenient as it limits the user's mobility. However, lot of hindrance have been faced while undergoing this research such as blur background, inconsistency, motion blur, etc. Sun et al. [4] developed a hand gesture detection system using a vision-based methodology. To address the issues of consistency, over fitting, information redundancy, and dimension disaster, they used the ANN, SVM, and KNN models. Three layers make up the ANN architecture of the recommended system: an input layer, a hidden layer, and an output layer. In their paper, the input layer is made up of 40 neurons, representing the 40 features, and the output layer

consists of 40 neurons, representing the 40 classes of gestures. The network was trained using a variety of hidden unit counts. This allowed the system to record the network topology entirely and properly. The weights were changed using the back-propagation process.

In [5], to detect hand movements, deep neural network topologies like CNN and LSTM network have been involved. For testing, a K-fold cross-validation technique was introduced. Bu [6] has used different motion videos to detect the gesture from these, and they have fused various neural features using convolutional neural network in order to achieve high accuracy. In [7], HOG is used to increase accuracy to characterize an object's appearance and shape in an image. It is computed on a large grid of cells, and adjustments are made to the contrast between the cells. Another CNN-based architecture was used to enable static hand recognition [8]. A model 32-by-32 pixel image that just contained the hand region was supplied to this architecture. A CNN and an LSTM were integrated in order to recognize temporal 3D pose movements from input sequences that accommodate the three dimensional poses of the hand gestures. A powerful mark-less hand gesture recognition system was developed that is capable of accurately detecting both static and dynamic hand gesturing movements. The best element is that it may immediately relate to real-time applications; for instance, the dynamic method can alter the slides in a presentation. Other applications include starting VLC player and opening Websites, among others.

The first real-time detection method is the Viola-Jones target detection approach which relies on Haar feature eradication [9]. Here, machine learning approaches are frequently used such as SVM, KNN, and decision trees for feature extraction. The deep learning-based 3D neural network model to recognize hand gesture opened the door for human computer interactions and the development of sign language utilizing transfer learning addresses the dearth of huge datasets [10]. Nunez et al. [11] described that sometimes, it can be difficult to recognize hand gestures against intricate backdrops. However, it is accomplished by employing the Kinect sensor to identify the dominating red, blue, and green—depth in brief RGB-D, followed by a gigantic fusion of two streams CNN constructing and training to produce the robust recognitions of hand gestures in complex backgrounds.

Haria et al. [12] presented the development of a robust marker-free hand gesture detection system that is capable of tracking both static and moving hand motions. Our technology converts the motion into behaviors like accessing Webpages. In [13], preprocessing steps like gamma correction was used for noise reduction and for enhancing the quality of the image. After that, thresholding-based segmentation was used to separate the foreground from background. The segmentation creates a binary image, which is then converted into a scaled, cropped image of a hand motion by performing dilation and erosion. Ameen and Vadera [14] suggested a parallel CNN using color image and a depth image. Here, RGB image and depth image input are fetched into the network to detect movements of the gestures. Here, a softmax classifier is also used with the last layer of CNN to achieve good accuracy.

### 3 Proposed Method

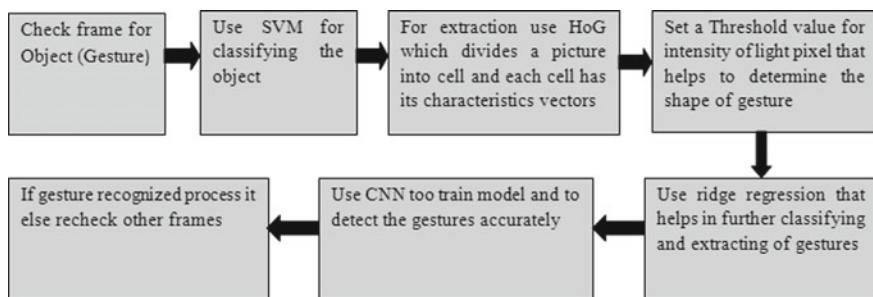
In our proposed method, the very first step is to determine whether the image consists of the object or not. Finding out where the image's target is located is the second criterion. This work introduces the histogram of oriented gradient (HOG) extraction technique along with the support vector machine (SVM) classification algorithm for detecting the hand gestures. SVM is a classification technique for supervised machine learning which determines the ideal boundary or hyperplane for partitioning the different classes of data. By locating the ideal boundary that splits the feature vectors of the different movements, SVM is trained to recognize a variety of hand gestures. To represent the particular object in its different shape, HOG is used. It can be represented as a gradient vector and can be identified by the intensity and the total distribution of pixel values. It also represents the set of components that together show how quickly the value of a pixel can change. The gradient vector value contains the information of insightful data. It also shows how the brightness value of pixels has changed. When a pixel is near an object's corner or edge, the gradient vector value changes. As a result, the HOG characteristic is useful for selecting the posture representation. The HOG approach aims to describe local objects in an image by using knowledge of the distribution of intensity gradients or edge directions. In order to implement the HOG operators, cells are used to segment a picture into smaller sections. We will create a histogram of the gradient's directions for each cell's points. The original image is represented by adding the histograms together. The HOG approach can extract specific elements from pictures that can precisely represent the shape and motion of the hand.

The second step is to locate the target when it is in motion or deformed in the following frames once the algorithm has identified the frame containing the object. This can be achieved by using support vector machine. It is primarily intended for quick development of perception that include reusable parts for inferencing. Furthermore, it makes it simpler to include computer vision software into programs and demonstrations that operate on a variety of hardware systems. Deep neural networks are its primary emphasis, yet it may be utilized for a variety of purposes. The SVM classifier can then correctly categorize the hand motion using these properties. Local histograms can be adjusted for contrast to improve recognition performance by determining an intensity threshold in a region greater than the cell calling blocks. The block's cells will all be normalized using the threshold value. A feature vector that is more resilient to variations in lighting will be the outcome of the normalization process.

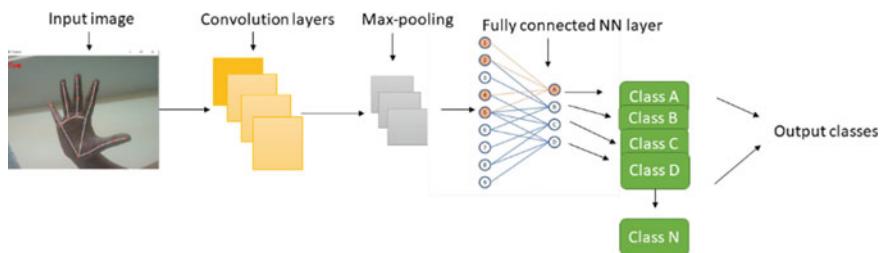
In third phase, neural network is used to identify the target. The human brain serves as the inspiration for neural networks. It resembles how organic neurons communicate with one another. Node layers, which include an input layer, one or more hidden layers, and an output layer, make up neural networks. Techniques for classifying objects are especially used in the processing of images. The ability to differentiate between gestures that start with the same letter is a prerequisite for classifiers in artificial intelligence applications. This will guarantee the acceptable

level of control precision because the required accuracy is high. Processing speed sometimes is not as important because classification doesn't happen continuously in every frame. Moreover, hand key points also play a crucial role in detecting the hand gestures.

The final step is to identify the target when the item (or gesture) contains them. Once gesture recognition has started, we follow the target while ending the gesture. Convolution neural network (CNN) model is the one we use to carry out the task. CNN is employed in a variety of issues, including natural language processing and image recognition. From importing necessary packages to initializing the models and moving on to reading of frames from webcam, the important step in gesture recognizing leads to detecting of hand key points and paving a path to recognize the hand moments. Each step is important to achieve gesture recognitions. The work flow diagram of the proposed method which is depicted in Figs. 1 and 2 represents the architecture of the proposed method.



**Fig. 1** Work flow diagram of the proposed method used to detect hands gesture



**Fig. 2** Architecture of the proposed method used to detect hands gesture

## 4 Result and Discussion

The acquisition of datasets is a vital and fundamental phase in any proposed strategy for gathering evidence that aids in assessing and constructing superior remedies. The dataset used in this paper is from the benchmark dataset NUS hand posture dataset [15, 16]. Images with excellent clarity and detail were acquired from NUS hand posture dataset. In order to assess the performance of the proposed strategy, a total of 2022 images are used for training, followed by validation. We separated the dataset into training and testing sets because machine learning models train the data to learn its characteristics, and test data is needed for validation. In our proposed method, 60% dataset acted as the training set, while the rest 40% was deployed as a test set. Here, a 60% split provides training data, and a 40% split provides enough test data to evaluate the performance of the model. With 40% of the data allocated for testing, we can assess the model's ability to generalize to new, unseen data. In our proposed method, different poses of hand gestures experimented on NUS hand posture datasets [15, 16] is represented in Fig. 4. The datasets [15, 16] were last accessed on 1st September, 2022.

The experiments are performed in a desktop having Ubuntu 18.04 LTS, CPU 3.40 GHz and 4 GB RAM. The experiments are being conducted using Python 3.9.

### 4.1 Performance Evaluation

For assessing the robustness of the proposed method, four evaluation metrics have been used [12–14]. These evaluation metrics are basically calculated from the predicted samples which are useful for classification purpose. The formulae of accuracy, recall, precision, and F1-score are represented in Eqs. (1), (2), (3), and (4), respectively. Here, TP represents true positive, TN stands for true negative, FP depicts false positive, and FN illustrates false negative. Based on the predicted samples as depicted in Table 1, four performance metrics are used which are accuracy, recall, precision, and F1-score.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

**Table 1** Comparative analysis of the proposed method with other existing approaches in terms of evaluation metrics

Method used	Accuracy (%)	Recall	Precision	F1-score
Image-based [2]	30.02	0.23	0.30	0.26
Multi-scale feature method [4]	65.01	0.65	0.65	0.65
Inference fusion-based [13]	76	0.72	0.74	0.72
Vision-based deep learning [7]	78.64	0.78	0.77	0.75
Proposed method	80	0.81	0.84	0.83

**Fig. 3** Confusion matrix

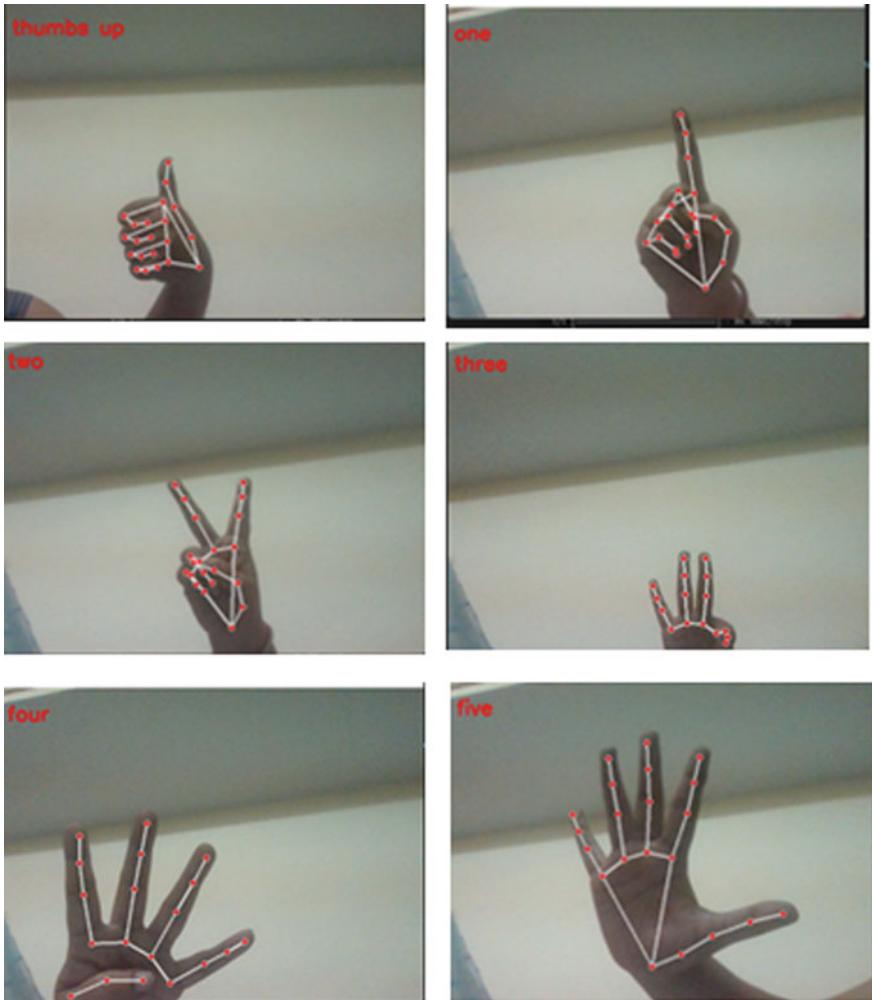
	A	B	C	D
Actual A	100	5	2	3
Actual B	3	95	1	1
Actual C	1	2	98	0
Actual D	2	1	0	97

The performance of a classification model, including hand gesture recognition, is frequently described using a confusion matrix, a table. The columns of the matrix reflect the predicted labels (or predicted values) of the samples, whereas the rows of the matrix represent the genuine labels (or actual values) of the samples. Figure 3 matrix demonstrates the system's overall success, with the majority of predictions coming true. In addition to accurately identifying 100 samples of gesture A as motion A, the system also correctly identified 95 samples of gesture B as gesture B, 98 samples of gesture C as gesture C, and 97 samples of gesture D as gesture D. The algorithm incorrectly detected several other gestures, including five samples of gesture A and three samples each of gesture B and C.

The proportion of samples that are accurately anticipated when compared to a particular database constitutes the accuracy. The specificity is a metric for how accurately true negatives were detected. High specificity means the system is more adept at identifying negative instances. FPR stands for false positive response. It determines the proportion of real samples tested in a group among all anticipated specimens in terms of accuracy. Recall is the indicator of the proportion of real positive findings that are true positive. A system with higher accuracy has a better possibility of classifying real data in an appropriate manner. Recall, in contrast, shows how sensitive the suggested model is. The results of computing the harmonic mean of the accuracy and recall generate the F1-score. F1-score achieves its optimum output at 1, while 0 is the poorest. With each of these parameters, MCC is also used to return a value between -1 and 1. The prediction may be assessed as follows:

coefficient of 1-perfect prediction.

coefficient of 0—no better than random prediction.

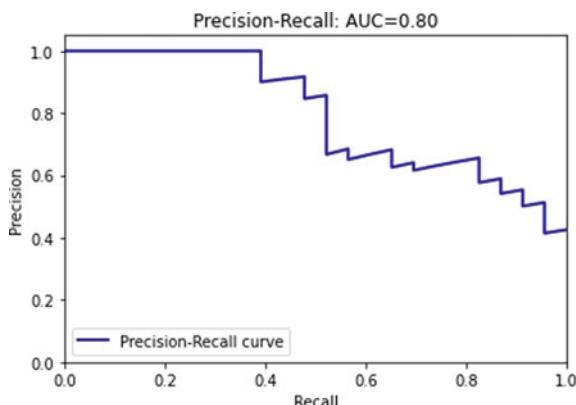


**Fig. 4** Different poses of hand gestures experimented on NUS hand posture dataset [15, 16]

coefficient of  $-1$ , no agreement between predicted and observed values. F1-score and area under receiver operating characteristics (AUC) both measure the accuracy of the model.

The graphical analysis of precision and recall of the proposed method is depicted in Fig. 5. Above graph represents the precision and recall of training and validation data. The comparative analysis of the proposed method has been performed with some existing approaches [2, 4, 7, 13], and the experimental outcomes demonstrate that the proposed approach shows its effectiveness in terms of an accuracy of 80%, precision of 0.84, recall of 0.81, and F1-score of 0.83 compared with other existing hand gesture recognition techniques.

**Fig. 5** Precision–recall analysis



## 5 Conclusion

This paper introduces an algorithm based on histogram of oriented gradient (HOG) and support vector machine (SVM) for detecting the hand gestures from raw RGB images. The computational overhead associated with recognizing hand posture using conventional methods is reduced by the proposed method, which does away with the need for hand identification and segmentation from the collected images. In addition, even with very slight interclass changes, the model can automatically derive the potential attributes that distinguish the hand postures. The proposed method's effectiveness has been assessed using K-fold cross validation on two publically accessible datasets [15, 16]. The performance of the proposed method is demonstrated by utilizing some statistical measures like accuracy, precision, recall, and F1-score. The proposed method can also automatically enhance its robustness by discriminating the hand movements even with very little interclass deviations. Hence, hand gesture recognition is a developing field with a wide range of potential applications, including virtual reality, human–computer interaction, and sign language recognition. The employment of cutting-edge methods like depth sensing, bone tracking, and machine learning algorithms has the potential to significantly increase the accuracy and reliability of hand gesture recognition systems, even though there are still issues to be resolved. In future, we would like to extend our work on detecting the movements of the hand from videos.

## References

1. Stergiopoulou E, Papamarkos N (2009) Hand gesture recognition using a neural network shape fitting technique. Eng Appl Artif Intell 22(8):1141–1158
2. Al-Hammadi M, Muhammad G, Abdul W, Alsulaiman M, Hossain MS (2020) Hand gesture recognition using 3D-CNN model. IEEE Consum Electron Mag 9(1):95–101

3. Zhan F (2019) Hand gesture recognition with convolution neural networks. In: Proceedings IEEE 20th international conference information reuse integration data science (IRI), July, pp pp 295–298
4. Sun Y et al (2020) Gesture recognition algorithm based on multi-scale feature fusion in RGB-D images. IET Image Process 14(15):3662–3668
5. Tran T-H, Do VH (2021) Improving continuous hand gesture detection and recognition from depth using convolutional neural networks. In: Intelligent systems and networks, Singapore, Springer, pp 80–86
6. Bu X (2020) Human motion gesture recognition algorithm in video based on convolutional neural features of training images. IEEE Access 8:160025–160039
7. Singha J, Roy A, Laskar RH (2018) Dynamic hand gesture recognition using vision-based approach for human–computer interaction. Neural Comput Applic 29:1129–1141
8. Huang J, Zhou W, Li H (2015) Sign language recognition using 3D convolutional neural networks. In: IEEE International conference on multimedia and expo (ICME), Turin, IEEE, pp 1–6
9. Viola P, Jones M (2004) Robust real-time face detection. Int J Comput Vision 57(2):137–154. <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>
10. Mohanty A, Rambhatla SS, Sahay RR (2014) Deep gesture: static hand gesture recognition using CNN. In: Raman B, Kumar S, Roy P, Sen D (eds) Proceedings of international conference on computer vision and image processing, Advances in intelligent systems and computing, vol 460. Springer, Singapore [https://doi.org/10.1007/978-981-10-2107-7\\_41](https://doi.org/10.1007/978-981-10-2107-7_41)
11. Núñez JC, Cabido R, Pantrigo JJ, Montemayor AS, Vélez JF (2018) Convolutional neural networks and long short-term memory for skeletonbased human activity and hand gesture recognition. Pattern Recognit 76:80–94
12. Haria A, Subramanian A, Asokkumar N, Poddar S, Nayak JS (2017) Hand gesture recognition for human computer interaction. Proc Comput Sci 115:367
13. Tao W, Leu MC, Yin Z (2018) American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion. Eng Appl Artif Intell 76:202–213
14. Ameen S, Vadera S (2016) A convolutional neural network to classify American sign language fingerspelling from depth and colour images. Wiley Expert Systems
15. <https://scholarbank.nus.edu.sg/handle/10635/137241>. (Accessed on 5 Jan 2023)
16. <https://www.ece.nus.edu.sg/stfpage/elepv/NUS-HandSet/>. (Accessed on 22 Dec 2022)

# A Combined Approach of Color Correction and Homomorphic Filtering for Enhancing Underwater Images



R. Ahila Priyadharshini and K. Ramajeyam

**Abstract** Underwater imaging is crucial for marine biology, oceanography, and underwater archeology, but low-quality images often result from issues like speckle noise, backscatter noise, and blur. To address the difficulties posed by these challenges, a recent study presents an effective methodology for enhancing the visual quality of underwater images. The approach comprises color correction, contrast enhancement, homomorphic filtering, and fusion. Contrast stretching is also used to improve contrast based on the range of intensity values. The proposed method is evaluated through qualitative and quantitative assessments, demonstrating its effectiveness in improving image details, enhancing global contrast, and exposing dark areas. According to the results, the proposed methodology surpasses several state-of-the-art techniques currently in use. The ultimate objective is to enhance the visual quality of underwater images, and the outcome of the study is a strong methodology that effectively tackles the challenges of underwater imaging, leading to improved image quality.

**Keywords** Underwater images · White balancing · Contrast enhancement · Homomorphic filtering · Fusion · Contrast stretching

## 1 Introduction

In recent years, the image processing and underwater vision fields have placed significant emphasis on enhancing the quality of underwater images. Both areas have recognized the importance of enhancing the clarity and visual representation of underwater imagery [1]. The development of underwater imaging systems for military drones [2], as well as the growing interest from the commercial and consumer sectors, has led to increased accessibility and exposure in various marketplaces [3]. However, there are fundamental constraints imposed by the underwater environment, which presently

---

R. Ahila Priyadharshini (✉) · K. Ramajeyam

Department of ECE, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu 626005, India  
e-mail: [rahila@mepcoeng.ac.in](mailto:rahila@mepcoeng.ac.in)



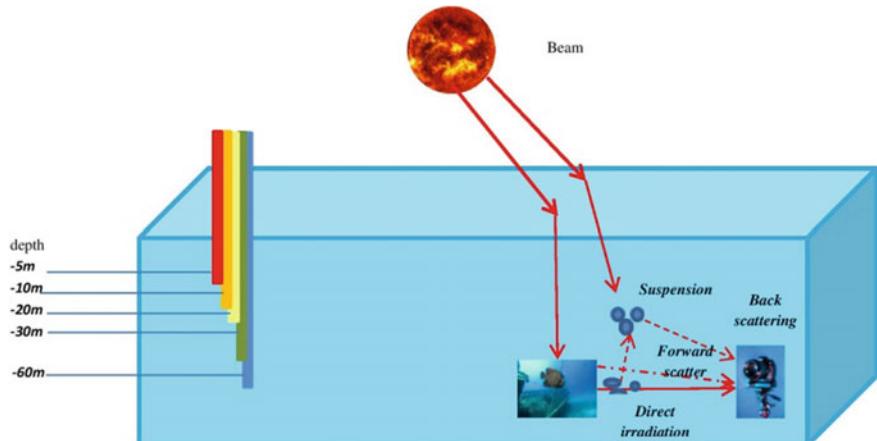
**Fig. 1** Examples of underwater images with different degradations

limit the quality of these images. The presence of speckle noise, backscatter noise, color distortion, motion blur, low visibility, and low contrast degrades the overall quality of the image [4]. These factors arise from the optical process of forming underwater images, which is distinct from that of imaging in the air due to light scattering in the underwater medium. Despite these challenges, it is essential to avoid over-enhancing and saturating underwater images. Figure 1 presents a collection of underwater images that exhibit different types of degradation.

According to the Jaffe–McGlamery model, which is shown in Fig. 2 and referenced in [5], an underwater image can be divided into three parts:  $L_d$ ,  $L_f$ , and  $L_b$ .  $L_d$  represents the direct reflection from the object, which is not scattered by the water, while  $L_f$  refers to the forward scattering light reflected by the object at a small angle, and  $L_b$  is the backscattered light that enters the camera without reflecting off the object. Since underwater images are affected by complex imaging models and poor lighting conditions that can degrade their visual quality, enhancing their quality is essential to convert low-quality images into high-quality ones. Equation (1), which corresponds to Jaffe–McGlamery's model, can be used to decompose an underwater image into its components, thereby aiding in the understanding and improvement of the image's quality.

$$L_1 = L_d + L_f + L_b \quad (1)$$

Researchers have extensively explored methods to enhance the clarity of underwater images by addressing the haziness present in them. Li et al. [6] introduced a method that utilizes smearing and color correction algorithms to tackle color and contrast challenges in underwater images. To counteract the noise resulting from the physical characteristics of the underwater medium, they employed a histogram equalization algorithm and bilateral filtering for enhancing contrast and correcting colors. In their study, Wang et al. [7] introduced a method for enhancing low-light underwater images. Their methodology involved obtaining a haze-free image, followed by enhancing the contrast, brightness, and overall visual quality through various



**Fig. 2** Jaffe–McGlamery’s model [5]

techniques such as histogram equalization, color correction, and stretching of the HSI color model. Priyadarshini and Aruna [8] proposed an effective technique for enhancing visibility in hazy conditions to prevent road image degradation. Their approach removes haze without introducing noise or artifacts, making it suitable for applications like autonomous driving and surveillance systems. De Vleeschouwer et al. [9] developed an algorithm that utilizes a fusion approach with two inputs, namely a color-corrected image and an image with improved contrast. The algorithm estimates these inputs to enhance the overall quality of the underwater image. Khan et al. [10] employed a wavelet-based fusion method in their investigation to address the color and contrast issues often encountered in underwater imagery. They incorporated techniques such as histogram stretching and CLAHE to enhance the contrast and rectify the color distortion that occurs due to low contrast and color attenuation typical in hazy images. As a result, their approach proved effective in enhancing the overall quality of underwater images. In their research, Cosmin et al. [11] presented a method to address the difficulties posed by underwater conditions, where they produced two inputs showing a color-corrected and contrast-enhanced rendition of the initial underwater image. This technique combines the two inputs to mitigate the presence of halos and color distortions, thus leading to a refined image with reduced noise and enhanced visibility. Jiang et al. [12] put forward a novel approach to address the issue of degraded underwater images, which involves using a target-oriented perceptual adversarial network that incorporates an adaptive fusion of latent aspects. This technique is designed to effectively remove degradation in underwater images while prioritizing the preservation of specific target features. Wang et al. [13] introduced an approach to improve underwater images using wavelet decomposition and multi-scale fusion. Since water’s suspended particles can affect the image quality and accuracy, color correction becomes necessary in the image

processing domain. Zuiderveld et al. [14] have proposed a contrast-limited adaptive histogram equalization (CLAHE) method as an alternative approach for image enhancement. CLAHE balances the target areas by merging the histograms of adjacent regions. As this method works based on the neighborhood regions rather than the whole image, it can effectively deal with non-uniform lighting, leaving the processed image with balanced illumination. In their research, Iqbal et al. [15] presented an unsupervised methodology for improving the quality of low-quality images using color correction. The proposed approach utilizes color-balancing contrast correction for the RGB color model, as well as contrast correction for the hue (H), saturation(S), and intensity (I) color model.

Khan et al. [16] introduced bi-histogram equalization methods for improving the contrast of digital images while preserving their natural appearance. While multi-histogram equalization methods can maintain image brightness and authenticity, they may sacrifice either contrast or brightness. However, these methods are more appropriate for images with sufficient lighting and may introduce halo and color distortions in images with poor lighting conditions. The researchers addressed this limitation by enhancing the image results through the elimination of poor lighting conditions.

The aim of this research is to enhance the quality of images taken underwater, and various techniques are available for achieving this goal, each with its own strengths and weaknesses. Existing advanced techniques rely on color correction and multi-image processing, but these methods have downsides such as longer processing times, reduced image contrast, and lower accuracy of restoration. To address these issues, a need for a new and effective approach to underwater imaging has been identified. A novel procedure has been developed for removing the haze that commonly appears in conventional camera underwater images. The essential steps of this proposed methodology are outlined below.

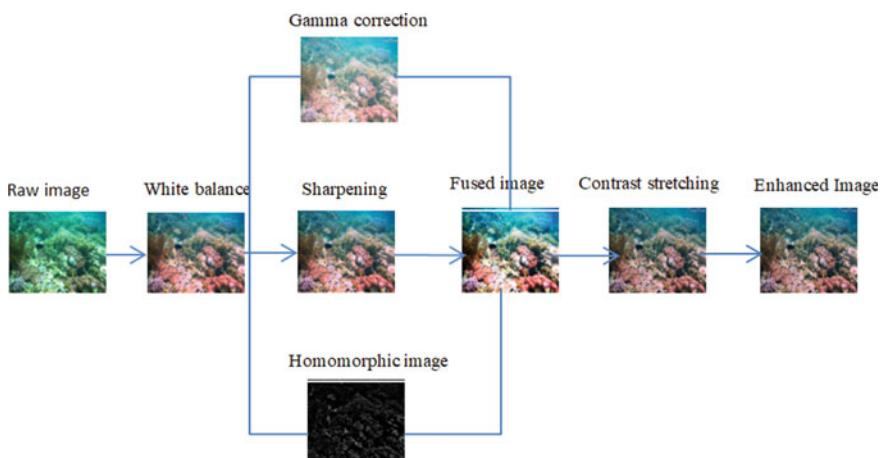
- To improve the quality of the reconstructed image, three algorithms—sharpening, homomorphic filtering, and gamma correction are used to create three variants of the original image.
- The three improved versions of the original image are then merged using a fusion technique that follows a maximum selection rule.
- The maximum selection rule selects the pixel with the highest intensity value from each corresponding input image to form the output image.
- After the fusion process, a contrast stretching algorithm is used to balance the dominant colors in the resulting image.
- The contrast stretching algorithm enhances the overall quality of the image.

## 2 Materials and Methods

The preceding section provided an overview of various studies conducted in the domain of underwater imaging. In this work, we have approached a new technique which utilizes a three-stage methodology to improve the quality of underwater images. This approach integrates white balance and image fusion techniques to produce superior quality images. The framework's design is shown in Fig. 3, which illustrates the different stages involved in the proposed strategy.

### 2.1 Color Balancing

Color balancing is a crucial step in enhancing underwater images as water can distort the colors due to light absorption at varying depths. The resulting images can appear too blue or green, leading to inaccurate and unappealing visuals. With color balancing, the natural colors of the scene can be restored, improving the overall aesthetic of the image. Techniques such as white balance adjustments, color correction tools, and color grading can be applied to adjust the colors of the image and create an accurate and visually pleasing final product. By applying these color-balancing techniques in underwater image enhancement, the unique beauty of the underwater world can be brought to life, enabling viewers to experience the natural colors of the scenery in their entire splendor.



**Fig. 3** Block diagram of proposed enhancement technique

### 2.1.1 White Balancing

White balancing [17] is a useful technique for eliminating undesired color casts from an image. This is especially important for underwater photography, where light scattering can cause color distortion. To achieve a natural look in underwater images, white balancing is typically applied to the three primary colors. However, in this environment, only the green color tends to be accurately preserved. As a result, to perform white balancing on underwater images, it is necessary to extract the red and blue channels using the green channel at each pixel location, applying Eqs. (2) and (3), and ensuring that the resulting normalized image rate falls within the range of 0 to 1.

For red color,  $I_{rc}$  at each pixel location ( $x$ ).

$$I_{rc}(x) = I_r(x) + \alpha[\overline{I_g} - \overline{I_r}][1 - I_r(x)].I_g(x) \quad (2)$$

For blue channel,  $I_{bc}$  at each pixel location ( $x$ ).

$$I_{bc}(x) = I_b(x) + \alpha[\overline{I_g} - \overline{I_b}][1 - I_b(x)].I_g(x) \quad (3)$$

### 2.1.2 Gamma Correction

Gamma correction [17] is a method used to adjust the contrast of an image between its dark and bright areas. This correction tends to reduce the contrast in underexposed regions while enhancing the darker areas of the image. By adjusting the luminance or brightness of an image, gamma correction aims to make it more consistent with the way the human eye perceives brightness [17]. This technique involves applying a specific function that maps the image's luminance levels to compensate for the nonlinear luminance effect that result from displaying an image on a screen.

### 2.1.3 Sharpening

In [17], the technique of normalized unsharp masking for image sharpening is explained. This method involves blending a blurred or unsharp version of an image with its original version to enhance its sharpness. The standard unsharp masking formula is utilized, which involves adding a certain parameter (denoted by  $\beta$ ) multiplied by the difference between the original image and its Gaussian-filtered version. The sharpened image  $I_s$  is obtained as  $I_s = I_{in} + \beta(I_{in} - G * I_{in})$ , where  $I_{in}$  is the image that needs sharpening (in this case, the white-balanced image),  $G * I_{in}$  represents the Gaussian-filtered version of the image  $I_{in}$ , and  $\beta$  is the parameter that needs to be determined. However, selecting an appropriate value for  $\beta$  is not an easy task. If  $\beta$  is too small, the sharpening will be ineffective, while if it is too large, the image

may have excessively bright highlights and dark shadows, leading to oversaturated regions. To address this issue, the sharpened image  $S$  is defined as given in Eq. (4).

$$S = (I + N\{I - G * I\})/2 \quad (4)$$

Here,  $N$  signifies the linear normalizing operator, commonly known as histogram stretching [17]. This method has the advantage of not requiring parameter adjustments and appears to be effective in terms of image sharpening.

## 2.2 Homomorphic Filter (HF)

The homomorphic filter [18] is designed on the basis of the assumption that an image can be modeled as a combination of two distinct components: an illumination component and a reflectance component. By applying the homomorphic filter, the illumination component is enhanced, while the reflectance component is attenuated, which ultimately leads to an improvement in image contrast and visibility. To achieve this, the homomorphic filter [18] utilizes a frequency-domain filtering process to separate the two components of the image. The illumination component represents the amount of incident light on the scene [18], while the reflectance component corresponds to the light reflected by the scene. Mathematically, for a given image  $m(x, y)$  at a pixel location  $(x, y)$ , the illumination component  $I(x, y)$  and the reflectance component  $R(x, y)$  are represented as given in Eq. (5).

$$m(x, y) = I(x, y) \times R(x, y) \quad (5)$$

In order to convert the image from the spatial domain to the frequency domain, a transformation function such as the Fourier transform is typically utilized [18]. Prior to this transformation, however, a logarithmic function is applied to Eq. (6), which involves changing the product of the illumination and reflectance components to the sum of the logarithmic components. This process is described as follows

$$Z(x, y) = \ln(m(x, y)) = \ln(I(x, y)) + \ln(R(x, y)) \quad (6)$$

Applying Fourier transform to Eq. (6) gives

$$Z(u, v) = F_l(u, v) + F_r(u, v) \quad (7)$$

A filter function  $H(x, y)$  is applied to the Fourier transformed signal, as shown in Eq. (7), which is then subjected to inverse Fourier transform to obtain the resulting function. An inverse exponential operation is performed on the resulting function to enhance the image.

### 2.3 Contrast Stretching

The contrast stretching technique, as described in reference [19], is used to enhance the contrast of an image by expanding the range of intensity values. This is achieved through the use of a linear scaling function, as given in Eq. (8), to adjust the pixel values. However, it should be noted that this method is only effective when the minimum and maximum intensity values of the image are different.

$$S = (r - r_{\min}) \frac{(I_{\max} - I_{\min})}{(I_{\max} - I_{\min})} + I_{\min} \quad (8)$$

Here  $r$  is used to represent the current pixel intensity value. The minimum intensity value in the entire image is denoted by  $r_{\min}$ , while  $r_{\max}$  is used to represent the highest intensity value within the image.

### 2.4 UIEB Dataset

The “UIEB dataset (underwater image enhancement benchmark),” as described in [20], includes 890 authentic underwater images that have been captured under different lighting conditions. The dataset also includes a corresponding reference image that has better visibility and brightness compared to the original image, and does not contain a color cast, providing a more authentic representation of the scene.

## 3 Experiments and Discussions

Our approach to enhancing underwater images comprises three distinct steps, which are input processing, white balancing, homomorphic filtering, and color correction, with the final step being image fusion. The fusion technique utilized in our method involved an average pixel-level approach that mitigates the effects of backscattering, ultimately resulting in superior quality images. In the following section, we assess the sensitivity and performance of the fusion technique proposed in this study regarding its ability to handle low-contrast effects. We compare our approach to existing methods such as “He et al. [21], Ancuti et al. [22],” and various underwater dehazing approaches including “Drews Jr. et al. [23], Galdran et al. [24], Emberton et al. [25], and Vleeschouwer [9].” We conducted an evaluation of ten methods, including our proposed approach, on the UIEB dataset. This publicly available dataset comprises images with various levels of color distortion, low resolution, and fog. Eight images were selected from the dataset’s validation set for our validation experiments, which were conducted using several non-reference evaluation metrics, such as UIQM [26] and UCIQE [26], to compare the experimental results.

### 3.1 Underwater Image Quality Evaluation Metric

The objective of the underwater image quality evaluation metrics is to analyze and evaluate the processed underwater image [26]. Currently, there are two established methods for evaluating the quality of such images, namely the “underwater color image quality evaluation (UCIQE) and underwater image quality metrics (UIQM).”

#### 3.1.1 Underwater Color Image Quality Evaluation (UCIQE)

In order to evaluate non-uniform color casts, blurring, and noise in underwater monitor images, the UCIQE method presented in Eq. (9) was utilized. The UCIQE approach involves converting an underwater image from RGB to CIELAB, which is more consistent with the human visual system, allowing for a more accurate measurement of these image quality metrics [26]. A higher UCIQE score suggests greater harmony between chroma, saturation, and contrast [26]. The calculation of UCIQE is expressed using Eq. (9):

$$\text{UCIQE} = c_1 \times \sigma_c + c_2 \times \text{con}_l + c_3 \times \mu_s \quad (9)$$

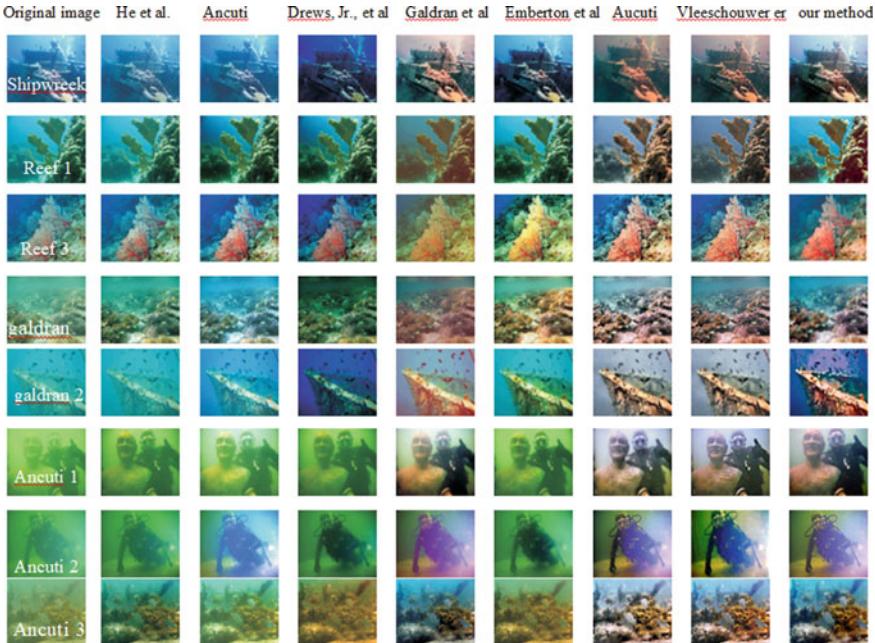
‘where  $\sigma_c$  is the chroma standard deviation,  $\text{con}_l$  is the luminance contrast, and  $\mu_s$  is the average saturation value. These are the weight coefficients:  $c_1$ ,  $c_2$ , and  $c_3$ ’

#### 3.1.2 Underwater Image Quality Metrics (UIQM)

The UIQM is a metric for evaluating the quality of underwater images, which is modeled after the way the human visual system functions [26]. The evaluation of the quality of underwater images is based on three unique metrics: colorfulness, sharpness, and contrast. These measures are specifically tailored to assess the quality of images captured in an underwater environment. A higher score on the UIQM metric indicates that the resulting image is more consistent with the human perception of quality [26]. The UIQM is based on a model that is designed specifically for underwater images and aims to accurately capture their unique characteristics. The calculation of UIQM is expressed using Eq. (10):

$$\text{UIQM} = c_1 \times \text{UICM} + c_2 \times \text{UISM} + c_3 \times \text{UIConM} \quad (10)$$

where  $\text{UICM}$ ,  $\text{UISM}$ , and  $\text{UIConM}$  correspond to the image colorfulness, sharpness, and image contrast measures, respectively, and  $c_1$ ,  $c_2$ ,  $c_3$  are the weight coefficients. This paper present a comparison of their proposed image enhancement method to several existing approaches for both outdoor and underwater dehazing. The outdoor methods include those developed by “He et al. [21] and Ancuti and Ancuti [22], while the underwater methods include those developed by Drews Jr. et al. [23], Galdran

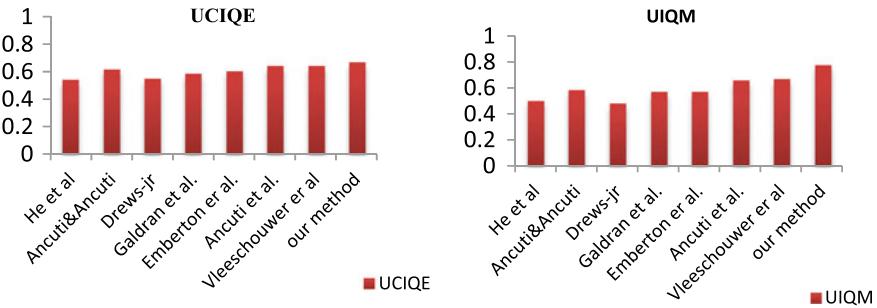


**Fig. 4** “Comparison to different outdoor (He et al. [21] and Ancuti and Ancuti [22]) and underwater dehazing approaches (Drews Jr et al. [23], Galdran et al. [24], Emberton et al. [25], Vleeschouwer [9], Ancuti et al. [9] and our proposed approach). The quantitative evaluation associated to these images is given in 1 and 2”

et al. [24]., Emberton et al. [25]., Vleeschouwer. [9]., Ancuti et al. [9]., and our proposed approach.” The comparison is presented in Fig. 4.

The quantitative evaluations of the different methods are provided in Tables 1 and 2. Based on these evaluations, it is claimed that the proposed method outperforms the existing methods in terms of objective image quality metrics such as UCIQE and UIQM. Additionally, visual comparisons of the results produced by the proposed method and the existing methods are provided, demonstrating that the proposed method is capable of producing clearer and more visually pleasing images. Overall, the comparison suggests that the proposed method is a promising approach for outdoor and underwater image enhancement.

The fusion-based strategy outperforms the dehazing algorithms of “Galdran et al. [24] and Emberton et al. [25].” in recovering visibility in considered scenes. However, according to Tables 1 and 2, the proposed strategy has similar or higher UCIQE and UIQM values than these algorithms. The proposed approach also produces good visual quality, with significant improvements in global contrast, color, and architectural features. Compared to the multi-scale methodology provided in [9], the proposed approach is more robust in harsh underwater environments, such as murky sea water and non-uniform artificial illumination. This is supported by Fig. 5, which shows the proposed algorithm producing clearer and brighter results with



**Fig. 5** Comparison of average image quality metrics by underwater images

more obvious color contrast compared to the fusion algorithm in difficult underwater situations.

Finally, the proposed methodology for “underwater image enhancement” has been found to have the highest level of color correction accuracy, with more diverse color intensity than state-of-the-art approaches. This was further confirmed by the discovery of a color cast, which was corrected more accurately by the proposed method than by the other approaches. The proposed method also offers a simplified version of the fusion method that can replace the mean pixel fusion, albeit at the cost of lower image detail quality. In Fig. 5, a comparison of the average image quality metrics for a set of underwater images is presented. This comparison was carried out using a proposed method for image fusion, which involves combining three image processing techniques (sharpening, gamma correction, and homomorphic filtering) with the corresponding pixel averages of images to create a fused image. Overall, the proposed method is effective in addressing various underwater image distortion scenarios.

## 4 Conclusion

This study presents a novel method for improving underwater images by using a fusion principle that follows a maximum selection rule. The proposed method does not require any additional information beyond the original image and effectively enhances various types of underwater images, including those affected by scattering, absorption, noise, haze, low contrast, and color distortion. The method accurately recovers essential faded features and edges, and Fig. 4 illustrates how it works.

Our algorithm provides an efficient and effective method for enhancing the quality of hazy images captured in various underwater environments. Based on the simulation results of UCIQE and UIQM, our fusion approach for improving underwater images outperforms earlier techniques. This technique successfully enhances the image quality while preserving crucial details and edges.

**Table 1** Comparison of UCIQE metrics in our method with other existing approaches (The best result is in bold)

Sample images	He et al. [25]	Ancuti and Ancuti [9]	Drews Jr [23]	Galdran et al.[24]	Emberton et al. [25]	Ancuti et al. [9]	Vleeschouwer et al [9]	Proposed method
Shipwreck	0.565	0.629	0.55	0.646	0.632	0.634	0.632	0.681
Reef 1	0.612	0.657	0.649	0.576	0.66	0.655	0.658	0.6725
Reef 3	0.606	0.661	0.62	0.533	0.678	<b>0.705</b>	0.697	0.6528
Galdran 1	0.593	0.631	0.544	0.529	0.652	0.643	0.659	0.6735
Galdran 2	0.426	0.558	0.536	0.596	0.63	0.667	0.633	0.694
Ancuti 1	0.485	0.561	0.499	0.641	0.499	0.588	0.594	0.621
Ancuti 2	0.456	0.595	0.492	0.529	0.529	0.59	0.592	0.6425
Ancuti 3	0.577	0.643	0.535	0.614	0.555	0.652	0.664	0.698

**Table 2** Comparison of UIQM metrics in our method with other existing approaches (The best result is in bold)

Sample images	He et al. [25]	Ancut and Ancuti [9]	Drews Jr [23]	Galdran et al.[24]	Emberton et al. [25]	Ancuti et al. [9]	Vleeschouwer et al [26]	Proposed method
Shipwreck	0.565	0.578	0.492	0.605	0.558	0.629	0.668	0.671
Reef 1	0.592	0.643	0.657	0.565	0.69	0.674	0.687	0.847
Reef 3	0.578	0.667	0.584	0.524	0.677	0.737	0.766	0.865
Galdran 1	0.578	0.601	0.519	0.569	0.664	0.669	0.68	0.832
Galdran 2	0.421	0.481	0.41	0.648	0.577	0.622	0.663	0.852
Ancuti 1	0.353	0.412	0.383	0.458	0.407	0.547	0.507	0.637
Ancuti 2	0.437	0.651	0.344	0.525	0.425	0.683	0.687	0.69
Ancuti 3	0.596	0.616	0.492	0.646	0.563	0.693	0.651	0.807

## References

1. Jaffe JS (2015) Underwater optical imaging: the past, the present, and the prospects. IEEE J Oceanic Eng 40(3):683–700. <https://doi.org/10.1109/joe.2014.2350751>
2. Prasath R, Kumaran T (2020) Application of different techniques for underwater image processing- a systematic review. In: IOP conference series: materials science and engineering, vol 925(1). pp 012034. <https://doi.org/10.1088/1757-899x/925/1/012034>
3. Strachan NJC (1993) Recognition of fish species by colour and shape. Image Vis Comput 11(1):2–10. [https://doi.org/10.1016/0262-8856\(93\)90027-e](https://doi.org/10.1016/0262-8856(93)90027-e)
4. Soni OK, Kumare JS (2020) A survey on underwater images enhancement techniques. In: 2020 IEEE 9th international conference on communication systems and network technologies (CSNT). <https://doi.org/10.1109/csnt48778.2020.9115732>
5. Jaffe JS (1990) Computer modeling and the design of optimal underwater imaging systems. IEEE J Oceanic Eng 15(2):101–111. <https://doi.org/10.1109/48.50695>

6. Li C, Guo J (2015) Underwater image enhancement by Dehazing and color correction. *J Electron Imaging* 24(3):033023. <https://doi.org/10.1117/1.jei.24.3.033023>
7. Li C-Y et al. (2016) Underwater image enhancement by Dehazing with minimum information loss and histogram distribution prior. *IEEE Trans Image Process* 25(12):5664–5677. <https://doi.org/10.1109/tip.2016.2612882>
8. Priyadarshini Ahila R, Aruna S (2018) Visibility enhancement technique for hazy scenes. In: 2018 4th International conference on electrical energy systems (ICEES). <https://doi.org/10.1109/icees.2018.8443201>
9. Ancuti CO et al. (2018) Color balance and fusion for underwater image enhancement. *IEEE Trans Image Process* 27(1):379–393. <https://doi.org/10.1109/tip.2017.2759252>
10. Khan A et al. (2016) Underwater image enhancement by wavelet based fusion. In: 2016 IEEE international conference on underwater system technology: theory and applications (USYS). <https://doi.org/10.1109/usys.2016.7893927>
11. Ancuti C et al. (2012) Enhancing underwater images and videos by fusion. In: 2012 IEEE conference on computer vision and pattern recognition. <https://doi.org/10.1109/cvpr.2012.6247661>
12. Jiang Z et al. (2022) Target oriented perceptual adversarial fusion network for underwater image enhancement. *IEEE Trans Circuits Syst Video Technol* 32(10):6584–6598. <https://doi.org/10.1109/tcsvt.2022.3174817>
13. Wang Y et al. (2017) Fusion-based underwater image enhancement by wavelet decomposition. In: 2017 IEEE International conference on industrial technology (ICIT). <https://doi.org/10.1109/icit.2017.7915500>
14. Zuiderweld K (1994) Contrast limited adaptive histogram equalization. In: Graphics gems, pp 474–485. <https://doi.org/10.1016/b978-0-12-336156-1.1.50061-6>
15. Iqba K et al. (2010) Enhancing the low quality images using unsupervised colour correction method. In: 2010 IEEE International conference on systems, man and cybernetics, 2010. <https://doi.org/10.1109/icsmc.2010.5642311>
16. Khan MF et al. (2012) Multi segment histogram equalization for brightness preserving contrast enhancement. Advances in intelligent and soft computing, pp 193–202. [https://doi.org/10.1007/978-3-642-30157-5\\_20](https://doi.org/10.1007/978-3-642-30157-5_20)
17. Tarhate S (2020) Weight maps guided underwater image enhancement by fusion technique. *Biosci Biotechnol Res Commun* 13(14):440–444. <https://doi.org/10.21786/bbrc/13.14/99>
18. Seow M-J, Asari VK (2004) Homomorphic processing system and ratio rule for color image enhancement. In: 2004 IEEE international joint conference on neural networks (IEEE Cat. No.04CH37541). <https://doi.org/10.1109/ijcnn.2004.1381031>
19. Khanal S (2020) Contrast stretching. Medium 5 June 2020. <https://samirkhanal35.medium.com/contrast-stretching-f25e7c4e8e33>
20. Li C et al. (2020) An underwater image enhancement benchmark dataset and beyond. *IEEE Trans Image Process* 29:4376–4389. <https://doi.org/10.1109/tip.2019.2955241>
21. He K et al (2011) Single image haze removal using dark channel prior. *IEEE Trans Pattern Anal Mach Intell* 33(12):2341–2353. <https://doi.org/10.1109/tpami.2010.168>
22. Ancuti CO, Ancuti C (2013) Single image Dehazing by multi-scale fusion. *IEEE Trans Image Process* 22(8):3271–3282. <https://doi.org/10.1109/tip.2013.2262284>
23. Drews Jr P et al. (2013) Transmission estimation in underwater single images. In: 2013 IEEE international conference on computer vision workshops. <https://doi.org/10.1109/iccvw.2013.113>
24. Galdran A et al. (2015) Automatic red-channel underwater image restoration. *J Visual Commun Image Representation* 26:132–145. <https://doi.org/10.1016/j.jvcir.2014.11.006>
25. Emberton S et al. (2015) Hierarchical rank-based veiling light estimation for underwater Dehazing. In: Proceedings of the British machine vision conference 2015. <https://doi.org/10.5244/c.29.125>
26. Guo P et al. (2022) Underwater image quality assessment: subjective and objective methods. *IEEE Trans Multimedia* 24:1980–1989. <https://doi.org/10.1109/tmm.2021.3074825>

# An Effective Pipeline for Depth Image-Based Hand Gesture Recognition



Taniya Sahana and Ayatullah Faruk Mollah

**Abstract** In this paper, a pipeline for hand gesture recognition from depth images is presented. This depth-based image recognition system is capable of recognizing gestures with challenges like varying depths, complex backgrounds, and variation in view point, hand pose, and appearance. Firstly, we obtain a grayscale image from the depth map, segment the hand region, and perform orientation normalization and feature extraction, which is followed by classification. Two different sets of feature descriptors are extracted: Multi-Radii Circular Signatures (MRCS) and Multi-Scale Density (MSD). Different classifiers have been used to demonstrate the efficacy of the suggested pipeline. Overall accuracy of 98.90% (MRCS) and 99.78% (MSD) is obtained using the MLP classifier.

**Keywords** Depth images · Hand segmentation · MRCS feature descriptor · Hand gesture recognition

## 1 Introduction

Hand gesture recognition creates a man–machine interaction system that is natural, easy to use, and easy to learn. The use of hand gestures in nonverbal communication is foremost. Thus, sign language, i.e., visual gestures and signs, provides a communication bridge for hearing and speech impairment community. Sign language is region-based; therefore, signs performed by the deaf–mute community are difficult for common people to understand. Different methods are used to recognize

---

T. Sahana

Department of Computer Science and Engineering, Dumkal Institute of Engineering and Technology, Basantapur, Murshidabad 742406, India

T. Sahana (✉) · A. F. Mollah

Department of Computer Science and Engineering, Aliah University, IIA/27 Newtown, Kolkata 700160, India

e-mail: [taniyaswork@gmail.com](mailto:taniyaswork@gmail.com)

A. F. Mollah

e-mail: [afmollah@aliah.ac.in](mailto:afmollah@aliah.ac.in)

hand signs. Each method has its own advantages and disadvantages. Sensors used in sign language recognition are either data gloves or video cameras [1]. In case of data gloves, user needs to wear data glove with some sensors attached [2]. Data gloves provide exact hand position and movement, but they hinder natural interaction. It prevents the natural motion of the hand because of the use of external hardware. Complex gestures can't be performed using this method. Most of these devices come at an expensive price. A video or vision-based method acquires hand gestures and motion. Hand gesture recognition is one of the fastest-growing research fields, having manifold challenges like locating hands, segmenting from complex backgrounds, varying lighting conditions, hand pose variations, intra-class variability, and changes in scale.

Use of depth-based images in hand gesture recognition has extensively increased with the introduction of different depth cameras or sensors like Microsoft Kinect [3], Creative Senz3D [4], Mesa Swiss-Ranger [5], etc. The new acquisition devices allow to get maximum information of hand shape in RGB, depth or videos. The benefit of using depth data in hand gesture recognition is that it is more robust to lighting changes, background, and noises [6]. Depth cameras produce a sequence of depth images that are later used for hand segmentation and tracking. Hand segmentation is easier for depth-based images as compared to images captured using color cameras. Many applications for depth-based hand gesture recognition systems have not yet been explored. Segmentation and feature extraction are challenging when images are taken in low illumination or complete darkness, or when an image contains a lot of debris, which appears as noise.

The primary goal of this paper is to develop a hand gesture recognition model for depth images. The method will be able to classify gesture samples that are even captured in low illumination or complete darkness. The proposed work is carried out using depth map data captured using a Creative Senz3D depth sensor. The remaining part of the paper is structured as follows: In Sect. 2, related work and motivation have been discussed. In Sect. 3, the segmentation scheme, including the limitations of single thresholding and the working principle of multi-level thresholding, has been presented; in Sect. 4, the results and analysis have been made; and finally, in Sect. 5, the conclusion and future works are outlined.

## 2 Related Work and Motivation

In the last two decades, there has been a lot of work conducted on recognizing hand gestures using images as well as video streams. Despite intensive research, recognizing hand gestures remained an open problem. According to the literature, vision-based recognition systems are classified into two types: static and dynamic. Without taking hand motion into consideration, static hand gestures try to categorize static images, also known as hand postures, into some stipulated classifications. In order to learn the motion of the hands, dynamic hand gestures are recognized using time-based data, i.e., hand detection and tracking [7]. Hand gesture recognition

typically follows these steps: image segmentation, preprocessing, feature extraction, and classification. Cao et al. [8] introduced a novel method to classify multi-class postures. Multiple image features were extracted, and multiple trained kernels of SVM were used. With the use of a publicly available dataset, Triesch et al. [9] achieved a recognition rate of 99.16%. On the other hand, Pisharady et al. [10] proposed a complete pipeline for hand detection, segmentation and recognition for multi-class postures in the presence of complex backgrounds. For the purpose of generating a saliency map for hand detection, the authors adopted a Bayesian model. Overall, 94.36% accuracy was obtained using multi-class SVM.

Research has also focused on contour-based hand sign recognition. Ren et al. [11] used hand contours for feature extraction and support vector machines for classification. For enhanced performance, they used a multi-scale weighted histogram and a contour direction-based normalization method. With a frame rate of 30 frames per second, they obtained a 97.10% accuracy rate. A real-time hand gesture recognition approach utilizing CNN was proposed by Zhan [12]. The dataset, which consists of 9 hand gestures and 500 images for each gesture, exhibited an average accuracy of 98.76%. A deep convolutional neural network technique for hand sign recognition was proposed by Adithya et al. [13]. They made extensive use of the vocabulary of sign language. They obtained 99.96% accuracy using two datasets that are freely available: the American Fingerspelling dataset and the NUS hand posture dataset.

Classical methods often fail to recognize hand gestures due to some inherent challenges. Hand detection and segmentation is difficult for images captured with complex backgrounds [14]. Geometrical variations that appear in the same gesture performed by different individuals are also a challenging factor [15]. Another significant challenge in automatic sign language recognition is the large number of classes with minimal inter-class variance [15]. Hand gestures acquired in varying illumination or in complete darkness appear difficult for segmentation and recognition [16].

The introduction of several depth sensors has created new possibilities for gesture recognition. Dominio et al. [17] proposed a novel hand recognition scheme for RGB-D images. Hand sign is mainly expressed in the palm and finger regions. Feature combination, i.e., distances of the fingertips from the palm center and curvature of the hand contour was taken. Overall accuracy of 99.50% was achieved using multi-class SVM. Wan et al. [18] proposed a work to recognize hand gesture captured using Kinect. Segmentation, feature extraction and matching, setting up 3D coordinates, trajectory extraction, and classification composed the recognition model. They achieved an average of 90% accuracy.

Though, hand gesture recognition using 3D model is becoming popular but less number of works in this area have been observed. Most of works on hand gesture recognition using depth-maps are found to employ Microsoft Kinect sensor. Wang et al. [19] and Yao et al. [20] have proposed different depth-based hand gesture recognition methods using Microsoft Kinect V1. Recent up-gradation of Microsoft Kinect V1 with different technologies inside have come up which is known as Microsoft Kinect V2, which has lead to the development of some other methods [21, 22]. Very few works have been reported using Creative Senz3D depth sensor. To carry out the

work, we have used publicly available Creative Senz3D dataset. This dataset contains gesture samples that are captured using Creative Senz3D camera under uneven lighting and complex background.

### 3 Depth Image Segmentation and Recognition

Segmentation is important to make objects of interest perceptible to machines. Image segmentation in machine learning splits and groups certain pixels together, actually assigning each pixel a label, and pixels with the same label fall under the same category where they have other things in common. A wrong segmentation weakens the classification process [23]. The depth image contains information related to the distance of the surface of the object from the camera. Illumination can be proportional to the distance from the camera or the nominal focal point. This makes nearer or farther surfaces darker or lighter. Most hand detection algorithms contend with identifying an exact hand shape when working with challenging backgrounds or varying lighting. Developing a universal hand segmentation technique is still a difficult task.

Though, single or bi-level thresholding is the simplest and known method for automatic image segmentation, the main drawback of single thresholding is that it assumes binary classifications. It divides the histogram of grayscale into two classes. However, segmentation issues typically arise in real-world scenarios when there are multiple classes or segments in the image. Single or other global thresholding methods do not give satisfactory results under these conditions: when noise is present in the image and there is a sharp valley between two peaks, the object of interest is smaller than the background, and the input image is taken with a complex background or captured in non-uniform illumination conditions. Segmentation of depth images using single or global thresholding does not give satisfactory results. In our proposed work, we have used a multi-level thresholding method to carry out the segmentation process. The proposed gesture recognition system consists of four main phases: hand segmentation, orientation correction, feature extraction, and classification.

#### 3.1 Hand Segmentation

An extended version of the single thresholding method is used. The method uses the mean and variance of pixel distribution. It calculates more than one threshold for a given input image. The input image is partitioned into different parts based on the threshold settings. Multiple thresholds generate multiple classes from a single input image [24]. If the gray-level intensity of an image is represented by  $(0, \dots, l - 1)$ , no. of pixels at level  $i$  is denoted by  $f_i$  then total no. of pixels at  $N = f_0 + f_1 + \dots + f_{l-1}$ . The occurrence of probability at level  $i$  is denoted by Eq. 1 [25].

$$p_i = f_i/N, \quad p_i \geq 0 \quad \sum_{i=0}^{l-1} p_i = 1 \quad (1)$$

If the image is divided into  $K$  different distinct regions or clusters, i.e.,  $C_0, C_1, \dots, C_{K-1}$ , then there are  $K-1$  threshold values, i.e.,  $t_0, t_1, \dots, t_{K-2}$ . Cumulative probability  $\omega_k$  and mean gray level  $\mu_k$  for each class is represented by Eqs. 2 and 3, respectively.

$$\omega_k = \sum_{i \in C_k} p_i \quad (2)$$

$$\mu_k = \sum_{i \in C_k} i \cdot p_i / \omega_k, \quad k \in \{0, 1, \dots, K-1\} \quad (3)$$

Equations 4 and 5 are used to calculate the image mean intensity  $\mu_T$  and the class variance  $\sigma_B^2$ .

$$\mu_T = \sum_{i=0}^{l-1} i \cdot p_i = \sum_{k=0}^{K-1} \mu_k \omega_k \quad (4)$$

$$\sigma_B^2 = \sum_{k=0}^{K-1} \omega_k (\mu_k - \mu_T)^2 = \sum_{k=0}^{K-1} \omega_k \mu_k^2 - \omega_k \mu_T^2 \quad (5)$$

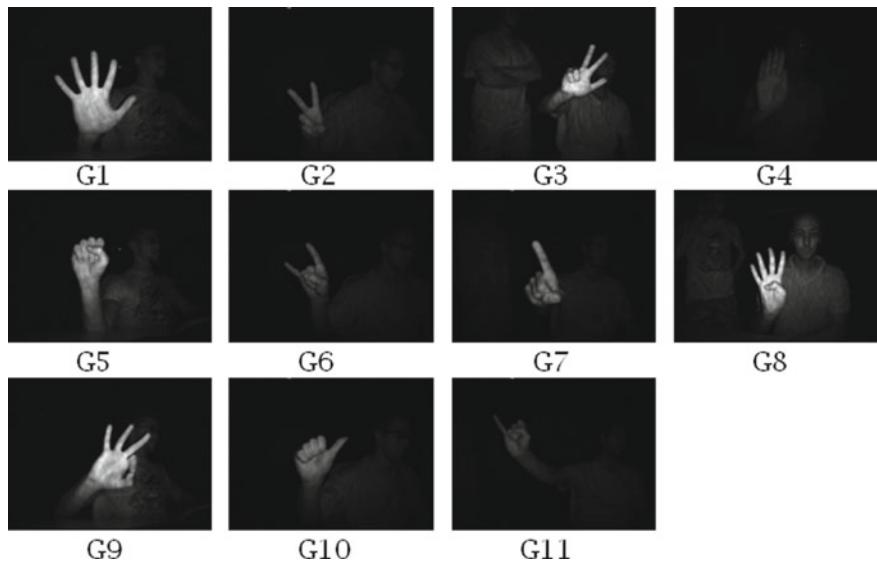
Maximization of class variance aids in determining the optimum thresholds  $(t_0^*, t_1^*, \dots, t_{K-2}^*)$  as shown in Eq. 6.

$$\{t_0^*, t_1^*, \dots, t_{K-2}^*\} = \arg \max_{0 \leq t_0 < t_1 < \dots < t_{K-2} < l-1} \{\sigma_B^2(t_0, t_1, \dots, t_{K-2})\} \quad (6)$$

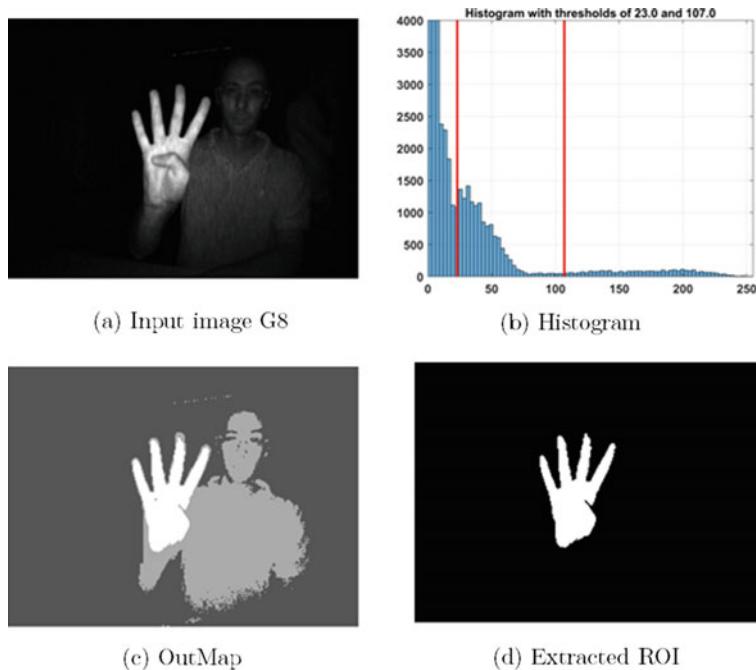
Here, the concept of single thresholding has been extended to K-level thresholding. Images containing two brightness regions are easier to segment. Maximization of between-class variance, or in other words, minimizing the within-class variance gives the optimal solution. A lower value of within-class variance means less dispersed data in the background and foreground. A maximum value of between-class variance means dispersion between two classes is highest. K-level thresholding is used for more complex images with different gray-level values.

In Fig. 1, different samples from the Creative Senz3D dataset have been shown. It is clear that this dataset contains gestures with different variations and orientations. Gestures with closest fingers are touching with each other.

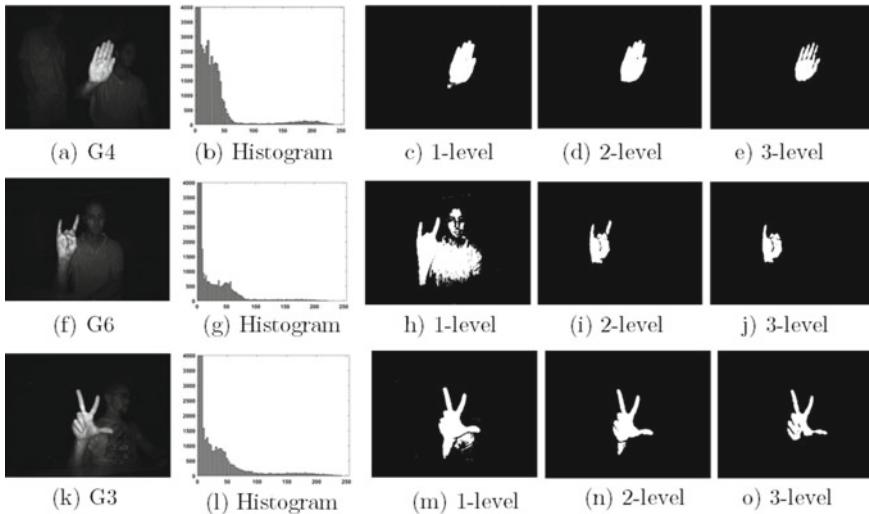
Figure 2 represents segmentation method that has been used in our work. Two threshold values are calculated for each input gesture. This is dividing the histogram into three distinct regions, hence dividing the image into three partitions or regions as well. Lower bound is set to 0 and upper bound is 255. All pixels within each range are set to an ID number. ID number is assigned in increasing order, i.e., the first region gets assigned an ID of 1, the next gets ID of 2, and so on. Each class is assigned a value that is a multiple of (256 divided by number of thresholds). Here, it is 64.



**Fig. 1** Gray-scale images with variations from Creative Senz3D dataset



**Fig. 2** Segmentation from a sample image. **a** Input gesture type G8, **b** histogram with thresholds, **c** out Map, **d** extracted hand ROI

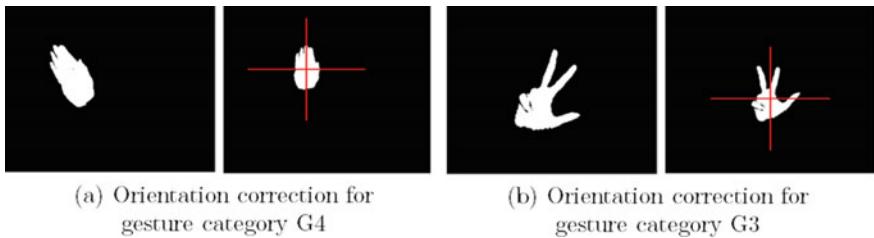


**Fig. 3** Segmentation using three different levels of thresholding for input gesture type G4, G6, and G3

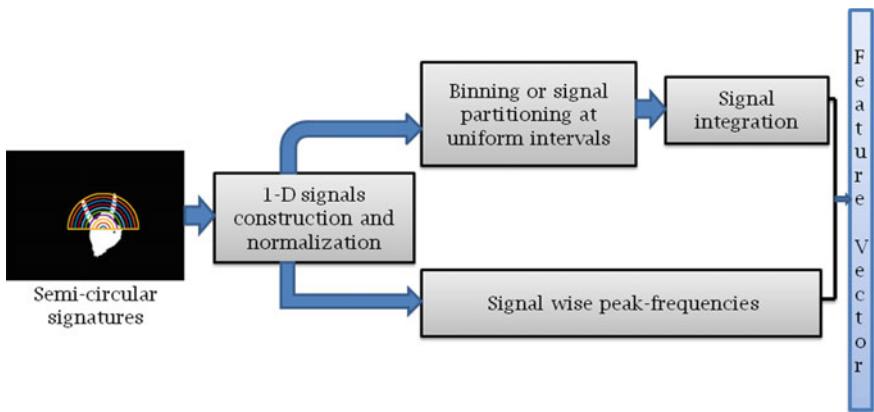
It may be noted that in this paper, we have focused on depth data, or a depth map, which is the distance of an object from the depth sensor. A depth image ordinarily has three levels: hand, body, and background. So the two-level thresholding reasonably justifies the segmentation procedure. In Fig. 3, gesture types G4, G6 and G3 have been segmented using different number of threshold values (i.e., 1, 2, and 3). Consideration of more than two thresholds may lead to oversegmentation. As a result, some portions may be chopped from the hand region, i.e., region of interest (ROI). In some other cases, noise may still be present when a single threshold value is considered. The histogram of the input image contains a slope between peaks that is taken in an uneven lighting condition. As the histogram is not bimodal, the single thresholding method is not able to produce satisfactory results. Noise can still be present in the segmented images. Basically, this method breaks two classes unequally, and  $\sigma_B^2$  may have two maximas. Hence, gestures get correctly segmented when two thresholds are applied.

### 3.2 Preprocessing

Hand segmentation is followed by orientation correction or normalization. Input symbols with different orientations are normalized to the pre-specified orientation, making the feature extraction and classification processes easier. For each input symbol, the Center of Gravity (COG) is determined [26, 27]. Orientation angle ( $\theta$ ) is calculated relative to the  $X$ -axis, and angle  $\phi = 90 - \theta$  is calculated in relation to  $Y$ -axis as well. Each input symbol is normalized by rotation of the angle  $\phi$ . Rotation



**Fig. 4** Orientation correction or normalization for gesture category **a** G4 and **b** G3



**Fig. 5** Steps followed in MRCS feature extraction

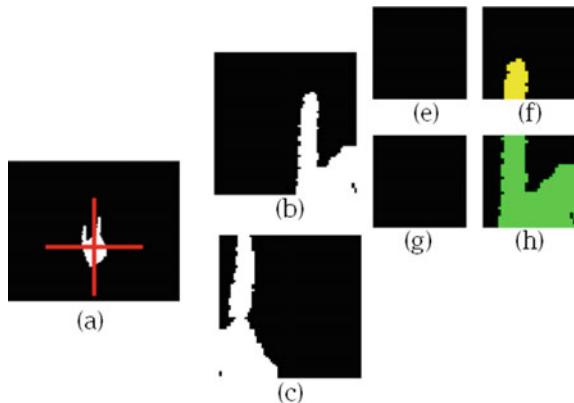
correction of two different sample gestures from G4 and G3 categories is shown in Fig. 4.

### 3.3 Feature Extraction

Two different features, namely Multi-Radii Circular Signatures (MRCS) [26] and Multi-Scale Density(MSD) [27], have been extracted after completion of the segmentation procedure. In MRCS, different radii of circular signatures are considered from the centroid of the hand region. These signatures are converted to one-dimensional binary signals and normalized within a specified range. Each binary signal is divided into certain bins, and the number of occurrences of 1's in each partition is counted. The number of peaks in each signal and binning make up the feature combination. The step-by-step MRCS feature extraction procedure for gesture type G6 is shown in Fig. 5.

In the case of MSD feature extraction, the input image is divided into four equal quadrants. Only the upper left and right quadrants are considered. The upper half of

**Fig. 6** Input image divided into subsequent smaller quadrants or images for multi-scale density feature extraction **a** input image, **b** upper left half, **c** upper right half, **d** upper left half first quadrant, **e** upper left half second quadrant, **f** upper left half third quadrant, and **g** upper left half fourth quadrant



the hand contains fingers and information that is required to build a recognition model. Then, the upper left and upper right halves are considered as two separate images, and each sub-image is further divided into four quadrants. This is a hierarchical approach. At each level of the hierarchy, each quadrant is considered a single image and is divided into four quadrants. This can go up to n-level. At each level of the hierarchy, a density feature (the total number of information pixels in each zone divided by the total number of pixels in the zone) is calculated.

A pictorial representation of zoning using a two-level hierarchy is shown in Fig. 6. Like the upper left half, the upper right half is also subsequently divided into four quadrants. Thus, the total number of obtained zones at the end of level two is 10.

## 4 Results and Analysis

### 4.1 Dataset Outline

Experiments have been carried out using the publicly available Creative Senz3D dataset [28, 29]. It contains static gestures captured using a Creative Senz3D camera. Thirty times each of 11 different static gestures are repeated by four different performers for a total of 1320 samples. Color, depth, and confidence frames are available for each sample. As a result, 3960 samples in total combined with confidence, depth, and RGB maps are acquired for each gesture. We have only used the depth map for our proposed work. The raw depth map and confidence map for Senz3D are each  $320 \times 240$  (short 16 bit) pixels, and the color map is  $640 \times 480$  pixels. Based on the location of the hand in front of the camera, numerous variants and orientations of gesture examples are collected. The mechanism utilized to determine the object's distance from the camera is the primary distinction between Senz3D and other depth cameras. Senz3D employs the "time-of-flight" approach to calculate

how long it takes light within its range of view to travel and reflect. This approach produces images with edges that are noticeably more tender than those from other depth sensors.

## ***4.2 Recognition Performance***

A 10-fold cross-validation framework has been used to test the efficiency of the proposed approach on the Creative Senz3D dataset. Multiple classifiers like Nave Bayes, KNN, SVM, RF, and MLP have been used to obtain the performance. In the case of the MRCS, nine, and for the MSD, two variants of feature combinations are used.

Some leading classifiers have been applied, and their parameters have been tuned. Naive Bayes is tuned with a supervised discretization function with batch size of 100; a linear search algorithm with batch size of 100 and  $k = 1$  has been used by KNN; SVM is tuned with batch size of 100, complexity constant of 1, seed of 1, tolerance parameter of 0.001, and poly kernel with catch size of 25007 and exponent 1.0. On the other hand, effective parameters for RF are batch size = 100, bag size percentage = 100 and number of iterations = 100. The highest accuracy has been achieved using the MLP classifier. MLP has used hidden layers = 2, no. of neurons in each hidden layer = 95, no. of epochs = 500 with learning rate = 0.3, momentum = 0.2. Table. 1 summarizes the obtained performance of each classifier against the MRCS and MSD features.

It may be noted from Table. 1 that recognition rates obtained with MSD are usually higher than that of MRCS for each classifier. In presence of challenging environment like varying illumination and complex background, poor contrast between foreground and background, satisfactory results have been obtained using the proposed method. The overall recognition rates have increased for the proposed method for each gesture category. Using the MLP classifier, we have a recognition rate of 98.90% for MRCS and 99.78% for the MSD. In Fig. 7, samples are taken in good contrast as well as in poor contrast. In both cases, proper hand shape is extracted using multi-level thresholding.

## ***4.3 Comparison with State of the art***

The use of multi-level thresholding over single thresholding has given promising results for most of the sample gestures. However, there are a few samples that have been misclassified. A closer look revealed that though these samples are captured in good lighting conditions, it is even impossible to extract the hand shape because the raw depth image does not look smooth and contains noise. Sometimes an extracted hand region either contains noise or a region of interest is improperly cropped, which

**Table 1** Performance evaluation of proposed method with MRCS and MSD features using different classifiers on Creative Senz3D dataset

Features used	No. of features	Naïve Bayes	KNN	SVM	RF	MLP
MRCS (signatures = 8, partitions = 8)	72	83.10	94.43	96.04	95.17	96.18
MRCS (signatures = 8, partitions = 12)	104	83.18	94.43	96.14	95.29	96.88
MRCS (signatures = 8, partitions = 16)	136	84.11	95.03	96.54	96.17	96.78
MRCS (signatures = 12, partitions = 8)	108	83.88	94.11	96.07	94.64	97.89
<b>MRCS(signatures = 12, partitions = 12)</b>	<b>156</b>	<b>86.01</b>	<b>96.18</b>	<b>96.30</b>	<b>97.55</b>	<b>98.90</b>
MRCS(signatures = 12, partitions = 16)	204	82.55	95.13	95.25	94.55	96.04
MRCS(signatures = 16, partitions = 8)	144	84.11	96.10	94.70	95.10	96.48
MRCS(signatures = 16, partitions = 12)	208	85.00	95.70	95.01	95.55	96.88
MRCS(signatures = 16, partitions = 16)	272	85.30	95.90	96.80	95.90	97.02
MSD(3rd level hierarchy, 42 zones)	42	93.33	95.52	98.40	97.88	99.67
<b>MSD(4th level hierarchy, 170 zones)</b>	<b>170</b>	<b>93.48</b>	<b>95.59</b>	<b>98.50</b>	<b>98.04</b>	<b>99.78</b>

Highest accuracy obtained has been marked as bold text

leads to failure in recognition. In Fig. 8, a few samples are shown where the multi-level thresholding method has failed to segment the hand region properly.

A comparative study of some other state-of-the-art methods and the proposed method on the Creative Senz3D dataset is given in Table 2. It is evident that by achieving higher recognition rates, it competes with state-of-the-art methods. In a real-world environment, it is difficult to build a recognition system in the presence of challenging environments. In computer vision, the majority of the state-of-the-art system uses sophisticated feature extraction and learning algorithms to yield robust performance. Automatic hand gesture recognition is still an open research area. Existing general purpose approaches are limited in their performance and robustness. We have tried to build a complete pipeline for hand gesture recognition from multiple shape cues with associated challenges.



(a) good contrast



(b) poor contrast

**Fig. 7** Segmentation of sample gestures in **a** good and **b** poor contrast**Fig. 8** Inaccurate segmentation and/or recognition of few gesture samples

**Table 2** Comparative study with some other state-of-the-art methods on Creative Senz3D dataset

Authors	Method	Recognition rate (%)
Memo et al.[28]	Depth thresholding and distance transform. Multi-class support SVM	90.00
Tang et al. [30]	Image Skeleton. Generative Adversarial Network	99.74
Liu et al. [31]	Image Skeleton. $\Delta$ -GAN	99.00
Miah et al. [32]	Multistage spatial attention based neural network. CNN	99.67
Bakheet et al. [7]	Morphological dilation and erosion. Support Vector Machine	93.30
<b>Proposed (MRCS)</b>	<b>Multi-level thresholding, Naïve Bayes, KNN, SMO, Random Forest, MLP</b>	<b>98.90</b>
<b>Proposed (MSD)</b>	<b>Multi-level thresholding, Naïve Bayes, KNN, SMO, Random Forest, MLP</b>	<b>99.78</b>

Highest accuracy obtained has been marked as bold text

## 5 Conclusion and Future Work

We proposed a pipeline for hand gesture recognition utilizing depth data in this paper. In this work, we have used a multi-level thresholding method that is able to segment images captured in uneven lighting and with complex backgrounds. The samples used in this paper are mainly multiple shape-oriented cues. Extracted feature descriptors are fed to different classifiers to perform gesture classification and recognition. In order to show the effectiveness of the proposed work, a 10-fold cross-validation framework has been used. Different classifiers like Nave Bayes, KNN, SVM, RF, and MLP have been applied. The highest accuracy of 99.78% is obtained in the case of MLP. However, the segmentation method used in this pipeline has only been applied to a single dataset. Two threshold values have been strictly considered for all the input symbols. In the future, one may consider other variants and challenges to make the method more general so that the method can be used to explore more challenging environments and more realistic datasets.

## References

1. Suarez J, Murphy RR (2012) Hand gesture recognition with depth images: a review. In: 2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication. IEEE, pp 411–417
2. Garg P, Aggarwal N, Sofat S (2009) Vision based hand gesture recognition. Int J Comput Inf Eng 3(1):186–191
3. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: CVPR 2011. IEEE, pp 1297–1304

4. She Y, Wang Q, Jia Y, Gu T, He Q, Yang B (2014) A real-time hand gesture recognition approach based on motion features of feature points. In: 2014 IEEE 17th international conference on computational science and engineering. IEEE, pp 1096–1102
5. Kapuscinski T, Oszust M, Wysocki M (2013) Recognition of signed dynamic expressions observed by tof camera. In: 2013 signal processing: algorithms, architectures, arrangements, and applications (SPA). IEEE, pp 291–296
6. Ali HH, Moftah HM, Youssif AA (2018) Depth-based human activity recognition: a comparative perspective study on feature extraction. *Future Comput Inf J* 3(1):51–67
7. Bakheet S, Al-Hamadi A (2021) Robust hand gesture recognition using multiple shape-oriented visual cues. *EURASIP J Image Video Process* 2021(1):1–18
8. Cao J, Yu S, Liu H, Li P (2016) Hand posture recognition based on heterogeneous features fusion of multiple kernels learning. *Multimedia Tools Appl* 75:11909–11928
9. Triesch J, Von Der Malsburg C (2001) A system for person-independent hand posture recognition against complex backgrounds. *IEEE Trans Pattern Anal Mach Intell* 23(12):1449–1453
10. Pisharady PK, Vadakkepat P, Loh AP (2013) Attention based detection and recognition of hand postures against complex backgrounds. *Int J Comput Vis* 101:403–419
11. Ren Y, Xie X, Li G, Wang Z (2016) Hand gesture recognition with multiscale weighted histogram of contour direction normalization for wearable applications. *IEEE Trans Circ Syst Video Technol* 28(2):364–377
12. Zhan F (2019) Hand gesture recognition with convolution neural networks. In: 2019 IEEE 20th international conference on information reuse and integration for data science (IRI). IEEE, pp 295–298
13. Adithya V, Rajesh R (2020) A deep convolutional neural network approach for static hand gesture recognition. *Procedia Comput Sci* 171:2353–2361
14. Stergiopoulou E, Sgouropoulos K, Nikolaou N, Papamarkos N, Mitianoudis N (2014) Real time hand detection in a complex background. *Engi Appl Artif Intell* 35:54–70
15. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. *Artif Intell Rev* 43:1–54
16. Muhammad H, Saud A, Shafiq A, Mazen Z, Shamsul H, Sofia I (2022) Hand gesture recognition with symmetric pattern under diverse illuminated conditions using artificial neural network. *Symmetry* 14(10):2045
17. Dominio F, Donadeo M, Marin G, Zanuttigh P, Cortelazzo GM (2013) Hand gesture recognition with depth data. In: Proceedings of the 4th ACM/IEEE international workshop on analysis and retrieval of tracked events and motion in imagery stream, pp 9–16
18. Wan T, Wang Y, Li J (2012) Hand gesture recognition system using depth data. In: 2012 2nd international conference on consumer electronics, communications and networks (CECNet). IEEE, pp 1063–1066
19. Wang C, Liu Z, Chan SC (2014) Superpixel-based hand gesture recognition with kinect depth camera. *IEEE Trans Multimedia* 17(1):29–39
20. Yao Y, Fu Y (2014) Contour model-based hand-gesture recognition using the kinect sensor. *IEEE Trans Circ Syst Video Technol* 24(11):1935–1944
21. Yang L, Longyu L, Dong H, Alelaiwi A, El Saddik A (2015) Evaluating and improving the depth accuracy of kinect for windows v2. *IEEE Sens J* 15(8):4275–4285
22. Gaber A, Faher MF, Waned MA (2015) Automated grading of facial paralysis using the kinect v2: a proof of concept study. In: 2015 international conference on virtual rehabilitation (ICVR). IEEE, pp 258–264
23. Abdulateef SK, Salman MD (2021) A comprehensive review of image segmentation techniques. *Iraqi J Electr Electr Eng* 17(2)
24. Liao PS, Chen TS, Chung PC et al (2001) A fast algorithm for multilevel thresholding. *J Inf Sci Eng* 17(5):713–727
25. Huang DY, Lin TW, Hu WC (2011) Automatic multilevel thresholding based on two-stage otsu's method with cluster determination by valley estimation. *Int J Innov Comput Inf Control* 7(10):5631–5644

26. Sahana T, Basu S, Nasipuri M, Mollah AF (2022) Mrcls: multi-radii circular signature based feature descriptor for hand gesture recognition. *Multimedia Tools Appl* 81(6):8539–8560
27. Sahana T, Paul S, Basu S, Mollah AF (2020) Hand sign recognition from depth images with multi-scale density features for deaf mute persons. *Procedia Comput Sci* 167:2043–2050
28. Memo A, Minto L, Zanuttigh P (2015) Exploiting silhouette descriptors and synthetic data for hand gesture recognition. *Smart Tools Apps Graph* 15–23
29. Memo A, Zanuttigh P (2018) Head-mounted gesture controlled interface for human-computer interaction. *Multimedia Tools Appl* 77(1):27–53
30. Tang H, Wang W, Xu D, Yan Y, Sebe N (2018) Gesturegan for hand gesture-to-gesture translation in the wild. In: Proceedings of the 26th ACM international conference on multimedia, pp 774–782
31. Liu Y, De Nadai M, Zen G, Sebe N, Lepri B (2019) Gesture-to-gesture translation in the wild via category-independent conditional maps. In: Proceedings of the 27th ACM international conference on multimedia, pp 1916–1924
32. Miah AS, Hasan MA, Shin J, Okuyama Y, Tomioka Y (2023) Multistage spatial attention-based neural network for hand gesture recognition. *Computers* 12(1):13

# A Coupled System for Simultaneous Image Despeckling and Segmentation



Ankit Kumar and Subit K. Jain

**Abstract** Real-world images are inherently degraded by noise and intensity heterogeneity, posing significant challenges for image segmentation. To tackle these issues, we proposed a new active contour model that can effectively and accurately segment images which are affected by speckle noise and heterogeneity in intensity. The variational formulation of the present model consists of two key components: image despeckling and segmentation terms. The segmentation term is guiding the initial contour toward precise target boundaries, while the despeckling term is designed to minimize the impact of noise. To check the efficiency of the present model, we apply it on natural and medical images that exhibit high noise levels and intensity heterogeneity. Moreover, we utilize various assessment parameters to assess the effectiveness of the newly presented model. The numerical experiments reveal that the present model outperforms most existing active contour models.

**Keywords** Partial differential equations · Speckle noise · Variational model · Active contour model · Image segmentation · Ultrasound imaging

## 1 Introduction

Image segmentation is a crucial task of image processing and computer vision, but it can be challenging due to inherent heterogeneity in the intensity and noise [1]. This difficulty has significant practical implications for the analysis of images. Numerous mathematical models have been presented for segmentation of images in the last few decades, among which active contour models [2–4] have gained more attention because of their efficiency, accuracy, and robustness. Image segmentation involves extracting a certain area or area of interest from an image, and the accuracy of the segmentation plays a crucial role in the entire performance of automated visual inspection systems. However, images are often degraded by various imaging anomalies,

---

A. Kumar (✉) · S. K. Jain

Department of Mathematics and Scientific Computing, National Institute of Technology Hamirpur, Himachal Pradesh 177005, India  
e-mail: [ankitkumar2675@gmail.com](mailto:ankitkumar2675@gmail.com)

and these anomalies are inevitable during image processing or acquisition and posing a significant challenge for image segmentation. Manual segmentation is also tedious, time-consuming, and prone to errors, which highlighting the need of computer-based mathematical algorithms.

Traditional active contour models for image segmentation comprise the snake model [2] and the level set model [3]. The snake model is designed to attract a two-dimensional curve toward the boundaries of the target under the influence of image characteristics. On the other hand, the level set model can extract complex objects by representing the object's contour using an implicit function in a high-dimensional space. However, the limited capture range of the snake technique and the time-intensive re-initialization process required by the level set model render them inadequate for practical implementation in the field. Furthermore, models such as geodesic active contour [5] and distance regularized level set evolution (DRLSE) [6] are sensitive to noise as they utilized gradient-based edge stop function to evolve the level set function (LSF). The Chan-Vese (CV) model [4] is another widely used active contour model that employs global region characteristics for target segmentation. Despite its robustness to noise, the CV model cannot efficiently segment images with heterogeneity in the intensity.

In order to mitigate the effects of intensity heterogeneity, various models have been introduced, including the local region-based model [7] and the bias correction model [8]. However, these models frequently suffer from sensitivity to noise and initialization. Recently, coupled models [9, 10] that perform both image denoising and segmentation have gained popularity. Unfortunately, these models do not consider the noise distribution, which can lead to errors in both denoising and segmentation. To address these issues, we introduce a coupled model for the segmentation and despeckling of images simultaneously.

In summary, this study presents the following contributions:

- A new coupled active contour model that can effectively address both noise and intensity heterogeneity.
- Despeckling term along with segmentation term designed to mitigate Rayleigh speckle noise frequently found in ultrasound images.
- The numerical outcomes on a variety of images, including clean, noisy, and ultrasound images, confirm the superior performance of the current model in comparison to existing models.

This article is structured as follows: Sect. 2 presents a new model with the help of partial differential equations. Section 3 presents the experimental outcomes. Section 4 offers the conclusion of this study.

## 2 Methodology

The primary aim of image segmentation in this study is to accurately locate object boundaries in the presence of noise and heterogeneity. In this study, we have proposed a novel model to address these challenges simultaneously. The proposed model draws

inspiration from the DRLSE model [6] and Jin-Yang model [11]. By incorporating an image despeckling term and a gradient-based image segmentation term, our model is able to efficiently mitigate various levels of noise from noisy images and locate object boundaries in inhomogeneous and noisy environments. The proposed energy functional is designed to optimize both the segmentation and denoising tasks jointly, which ensures accurate and robust object boundary extraction. The proposed energy functional in the variational form is given by

$$\begin{aligned} E^{\text{Proposed}}(\varphi, I) = & \alpha \int_D g(|\nabla I|) \delta_\varepsilon(\varphi) |\nabla \varphi| dx + \beta \int_D g(|\nabla I|) \mathcal{H}_\varepsilon(-\varphi) dx \\ & + \frac{\gamma}{2} \int_D (|\nabla \varphi| - 1)^2 dx + \lambda \int_D \frac{(I_0 - I)^2}{I} dx + \int_D |\nabla I| dx, \end{aligned} \quad (1)$$

where  $\varphi$ ,  $I_0$ ,  $I$ , and  $D$  are the LSF, observed image, restored image, and image domain, respectively. Further, the edge function  $g(\cdot)$ , regularized Heaviside function  $\mathcal{H}_\varepsilon(\cdot)$ , and regularized Dirac delta function  $\delta_\varepsilon(\cdot)$  are defined in Eqs. (2)–(4) respectively.

$$g(\zeta) = \frac{1}{\zeta^2 + 1}. \quad (2)$$

$$\mathcal{H}_\varepsilon(\zeta) = \frac{1}{\pi} \left[ \frac{\pi}{2} + \arctan \left( \frac{\zeta}{\varepsilon} \right) \right]. \quad (3)$$

$$\delta_\varepsilon(\zeta) = \frac{1}{\pi} \frac{\varepsilon}{\varepsilon^2 + \zeta^2}. \quad (4)$$

The proposed model comprises five terms, each of which plays a distinct role in the contour evolution process. The first term acts as a guiding force, directing the curve toward the edges of the target. The second term maintains the direction of the curve, ensuring that it follows the target boundaries. The third term serves as a regularizer, promoting smoothness, and preventing contour leakage. The fourth term is the data fidelity term, which aims to suppress the speckle noise in the image. Finally, the fifth term acts as an image smoothing term, helping to eliminate image artifacts and further improving the segmentation results. The current study possesses the capability to perform image segmentation and despeckling simultaneously, resulting in the elimination of speckle noise and effective image segmentation even in inhomogeneous environments.

The Euler–Lagrange equation and gradient descent method are employed to minimize the Eq. (1). In this process, considering all parameters are constant, and minimizing the Eq. (1) with respect to  $\varphi$  and  $I$ , we get

$$\varphi_t = \alpha \delta_\varepsilon \operatorname{div} \left( g \frac{\nabla \varphi}{|\nabla \varphi|} \right) + \beta \delta_\varepsilon g + \gamma \left( \nabla^2 \varphi - \operatorname{div} \left( \frac{\nabla \varphi}{|\nabla \varphi|} \right) \right), \quad \text{in } D_T, \quad (5a)$$

$$I_t = \operatorname{div} \left( \frac{\nabla I}{|\nabla I|} \right) + \lambda \left( \frac{I_0^2 - I^2}{I^2} \right), \quad \text{in } D_T, \quad (5b)$$

$$\partial_n \varphi = 0, \quad \partial_n I = 0, \quad \text{on } \partial D_T, \quad (5c)$$

$$\varphi(x, 0) = \varphi_0(x), \quad I(x, 0) = I_0(x), \quad \text{in } D, \quad (5d)$$

where  $D_T := D \times (0, T)$ ,  $\partial D_T := \partial D \times (0, T)$ , and  $\partial_n$  denotes the derivative with respect to  $n$  at the boundary surface  $\partial D$  in the outward normal direction  $n$ . Additionally, Eqs. (5c) and (5d) depict the boundary and initial conditions of the proposed model.

In order to solve the presented model, the finite difference method has been employed. Explicit schemes are the simplest option and generally need smaller time steps to ensure stability. In this study, the explicit scheme with small time step ( $\Delta t = 0.2$ ) is employed to show the effectiveness of the current model. Therefore, the difference scheme that we have used is as follows:

$$\varphi_{i,j}^k = \varphi_{i,j}^{k-1} + \Delta t S_1(\varphi_{i,j}^{k-1}), \quad (6a)$$

$$I_{i,j}^k = I_{i,j}^{k-1} + \Delta t S_2(I_{i,j}^{k-1}), \quad (6b)$$

where  $S_1$  and  $S_2$  in Eqs. (6a) and (6b) are the discretized forms of right hand side of Eqs. (5a) and (5b), respectively. Furthermore, considered values of parameters in this study are as follows:  $\alpha = 5$ ,  $\beta \in [-2, 2]$ ,  $\gamma \in (0, 0.3)$ ,  $\lambda = 0.1$ ,  $\varepsilon = 1.5$ , and maximum number of iterations = 1500.

### 3 Results and Discussions

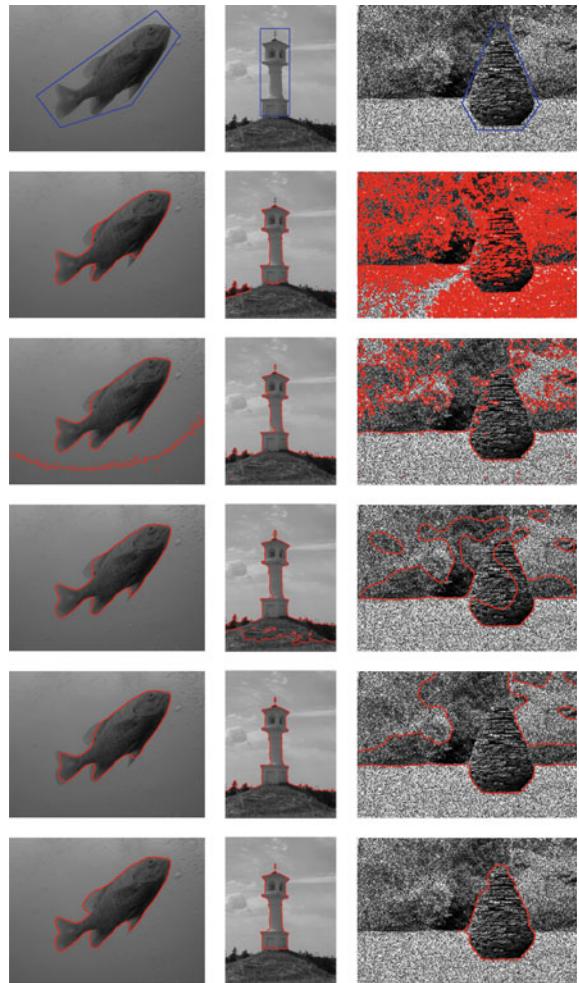
In this section, we illustrate an evaluation of the current model by comparing its results with other existing models in terms of assessment parameters considered in Table 1. MATLAB (R2020b) was employed to conduct all experiments on a Windows 10 (64bit) laptop featuring an Intel Core i5 processor with a processing speed of 1.60 GHz and a memory capacity of 8GB RAM.

In Table 1,  $S$  and  $G$  represent the segmented curve and gold truth curve, respectively. In addition, true positive ( $TP$ ) describes the pixels in the region of interest that are correctly labeled, while true negative ( $TN$ ) describes the non-region of interest pixels that are correctly labeled. False positive ( $FP$ ) indicates non-region of interest pixels that are mistakenly labeled as region of interest, and false negative ( $FN$ ) indicates the region of interest pixels that are mistakenly labeled as non-region of interest.

**Table 1** Assessment parameters for evaluating image segmentation

Assessment parameter	Symbol	Definition
Hausdorff distance [12]	HD	$\max_{c \in S} \min_{d \in G} \ c - d\ , \max_{d \in S} \min_{c \in G} \ c - d\ \}$
Dice coefficient [12]	DC	$\frac{2TP}{2TP+FN+FP}$
Accuracy [12]	A	$\frac{TP+TN+FP+TP}{TP+TN}$
Global consistency error [13]	GCE	$\frac{1}{ D } \min \left\{ \frac{FN(FN+2TP)}{FN+TP}, \frac{FP(FP+2TN)}{FP+TN}, \frac{FP(FP+2TP)}{FP+TP}, \frac{FN(FN+2TN)}{FN+TN} \right\}$
Matthews correlation coefficient [14]	MCC	$\frac{\sqrt{(FN+TN).(FP+TN).(FN+TP).(FP+TP)}}{TP+FP+FN+TN}$
Jaccard index [12]	IJ	
Sensitivity [12]	$\hat{S}$	$\frac{TP}{TP+FN}$

**Fig. 1** Segmentation outcomes for natural images. Left to right depicts Image 1 to Image 3, respectively. From top to bottom: images with their respective initial contour, segmented outcomes of ARB [15], HLFRA [16], AVLSM [9], RVLSM [10], and proposed model, respectively



### 3.1 Segmentation Results for Natural Images

For experiments, we selected two standard clean images and one noisy image from the dataset [17], that was degraded with Rayleigh speckle noise having a scale parameter of 0.6. For comparison, we evaluated four well-known models, namely Ali-Rada-Badshah model (ARB) [15], hybrid and local fuzzy region-edge-based active contour model (HLFRA) [16], adaptive variational level set model (AVLSM) [9], and robust variational level set method (RVLSM) [10], that are specifically designed to handle speckle noise. The parameter values used in these models are taken from their published implementations.

The first column of Fig. 1 shows that most of the models, except the proposed model, could not accurately locate the target boundaries because of weak edges. Similarly, these models performed poorly in the second column due to the cluttered background. However, the proposed model effectively located the object boundaries in the image with a complex background and high noise, while the other models were not able to handle this noisy image effectively. The experimental outcomes indicate that the present model is highly effective in handling images with diverse geometric regions, both with and without noise, thereby demonstrating its superiority over existing models.

### 3.2 Segmentation Results for Ultrasound Images

To further assess the effectiveness of the current model, we conducted experiments on three ultrasound images taken from the dataset [18]. As observed in Fig. 2, the proposed model was successful in accurately identifying and outlining the defective area in the ultrasound images. In contrast, other existing models were unable to handle the challenging characteristics of images, such as speckle noise and severe intensity heterogeneity.

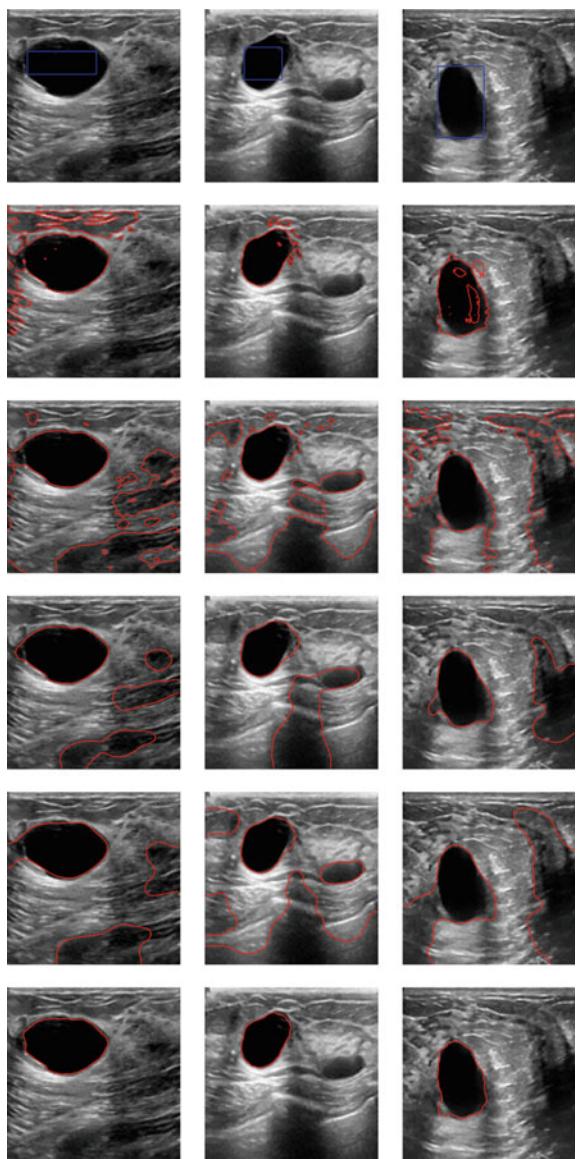
The quantitative outcomes for all the models are presented in Table 2 and Fig. 3. The present model exhibits the highest dice coefficient, Jaccard index, Matthews correlation coefficient, sensitivity and accuracy values, and the lowest Hausdorff distance and global consistency error values among the five algorithms for different images. This suggests that the current model is the most efficient in the form of considered evaluation parameters. Hence, the quantitative outcomes presented in Table 2 and the boxplot in Fig. 3 provide strong evidence to support the claim that the current model outperforms the existing models considered in this study.

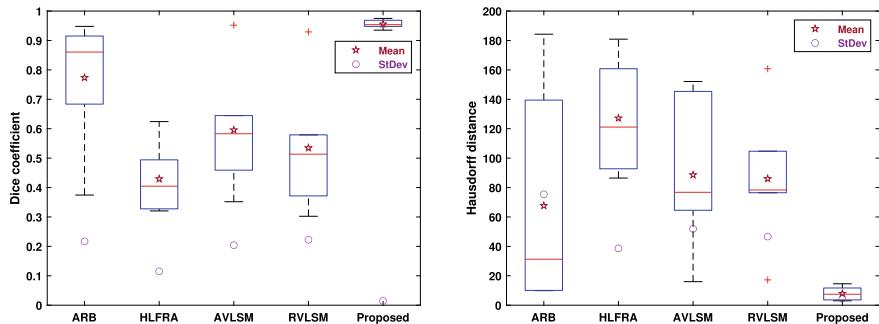
Finally, in terms of qualitative assessment, the present model effectively delineates target boundaries in both Figs. 1 and 2, surpassing the inefficiencies exhibited by the other models considered in this study for various imaging scenarios. Therefore, the proposed model exhibits superior performance in the qualitative analysis compared to the other models. Additionally, the quantitative analysis reveals that the present model achieves optimal segmentation measure values for the majority of images presented in Table 2 and Fig. 3. The precision of the proposed model is highlighted by the small box size, indicating the robustness of the model across different imaging scenarios. Hence, the proposed model demonstrates its effectiveness across a wide range of images.

**Table 2** Quantitative assessment of various evaluation parameters

Measure	Images	ARB [15]	HLFRA [16]	AVLSM [9]	RVLSM [10]	Proposed model
Accuracy [12]	Image1	0.9828	0.8068	0.9841	0.9770	0.9915
	Image2	0.9799	0.7740	0.8793	0.7804	0.9910
	Image3	0.6455	0.6006	0.7491	0.8294	0.9918
	Image4	0.8860	0.7566	0.8667	0.8294	0.9871
	Image5	0.9877	0.7261	0.8410	0.6910	0.9943
	Image6	0.9682	0.6248	0.8665	0.7001	0.9879
GCE [13]	Image1	0.0327	0.2909	0.0304	0.0429	0.0165
	Image2	0.0389	0.2970	0.1760	0.2877	0.0171
	Image3	0.4577	0.4990	0.3577	0.2497	0.0162
	Image4	0.1775	0.3323	0.2030	0.2497	0.0246
	Image5	0.0234	0.3299	0.2080	0.3662	0.0109
	Image6	0.0598	0.4474	0.1874	0.3688	0.0232
Jaccard [12]	Image1	0.9010	0.4537	0.9084	0.8677	0.9513
	Image2	0.8089	0.2683	0.4253	0.2884	0.9049
	Image3	0.2304	0.2394	0.2134	0.4076	0.9392
	Image4	0.5193	0.3282	0.4756	0.4076	0.9008
	Image5	0.8435	0.1959	0.2978	0.1781	0.9183
	Image6	0.7048	0.1909	0.3977	0.2282	0.8781
MCC [14]	Image1	0.9395	0.5646	0.9440	0.9188	0.9704
	Image2	0.8833	0.4103	0.5931	0.4496	0.9466
	Image3	0.2958	0.3451	0.2358	0.5447	0.9640
	Image4	0.6598	0.4583	0.6171	0.5447	0.9419
	Image5	0.9097	0.3607	0.4899	0.3346	0.9552
	Image6	0.8098	0.3322	0.5794	0.3894	0.9296
Sensitivity [12]	Image1	0.9997	0.4712	0.9996	0.9997	0.9996
	Image2	0.8866	0.2786	0.4357	0.2934	0.9997
	Image3	0.2430	0.2413	0.2648	0.4251	0.9595
	Image4	0.5329	0.3378	0.4917	0.4251	0.9953
	Image5	0.8740	0.1973	0.3002	0.1792	0.9952
	Image6	0.8036	0.1911	0.3996	0.2285	0.8934

**Fig. 2** Segmentation outcomes for ultrasound images. Left to right depicts Image 4 to Image 6, respectively. From top to bottom: images with their respective initial contour, segmented outcomes of ARB [15], HLFRA [16], AVLSM [9], RVLSM [10], and proposed model, respectively





**Fig. 3** Quantitative outcomes of considered models in terms of DC and HD. Mean and StDev describe the mean and standard deviation of each box, respectively

## 4 Conclusion

This study presents a new coupled active contour model designed for image segmentation with speckle noise and intensity heterogeneity. The proposed model comprises variational-based image segmentation and despeckling terms. The denoising term effectively removes various noise levels from noisy images, thus improving segmentation accuracy. The image segmentation term effectively locates object boundaries in noisy and inhomogeneous environments. We conducted comparison experiments on several types of images, including clean, noisy, and ultrasound images to check the proposed model's performance. The outcomes of the experiments demonstrate the efficiency and superiority of the current model over most existing models in successfully extracting objects from natural and medical images with speckle noise and intensity heterogeneity. In conclusion, the current model can significantly enhance the accuracy of image segmentation in challenging imaging scenarios. However, in the case of medical images characterized by severe intensity heterogeneity, the present study exhibits sensitivity to the initial contour. This intriguing challenge will be addressed in future research by integrating deep learning techniques into the present formulation.

## References

1. Kumar A, Jain SK (2022) Deformable models for image segmentation: a critical review of achievements and future challenges. *Comput Math Appl* 119:288–311
2. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. *Int J Comput Vis* 1(4):321–331
3. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys* 79(1):12–49
4. Chan TF, Vese LA (2001) Active contours without edges. *IEEE Trans Image Process* 10(2):266–277
5. Caselles V, Kimmel R, Sapiro G (1997) Geodesic active contours. *Int J Comput Vis* 22(1):61–79

6. Li C, Xu C, Gui C, Fox MD (2010) Distance regularized level set evolution and its application to image segmentation. *IEEE Trans Image Process* 19(12):3243–3254
7. Li C, Kao C-Y, Gore JC, Ding Z (2008) Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans Image Process* 17(10):1940–1949
8. Li C, Huang R, Ding Z, Gatenby JC, Metaxas DN, Gore JC (2011) A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI. *IEEE Trans Image Process* 20(7):2007–2016
9. Cai Q, Qian Y, Zhou S, Li J, Yang Y-H, Wu F, Zhang D (2021) Avlsm: adaptive variational level set model for image segmentation in the presence of severe intensity inhomogeneity and high noise. *IEEE Trans Image Process* 31:43–57
10. Zhang F, Liu H, Cao C, Cai Q, Zhang D (2022) Rvlsm: robust variational level set method for image segmentation with intensity inhomogeneity and high noise. *Inf Sci*
11. Jin Z, Yang X (2011) A variational model to remove the multiplicative noise in ultrasound images. *J Math Imaging Vis* 39(1):62–74
12. Karunananayake N, Aimmanee P, Lohitvisate W, Makhanov SS (2020) Particle method for segmentation of breast tumors in ultrasound images. *Math Comput Simul* 170:257–284
13. Taha AA, Hanbury A (2015) Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC Med Imaging* 15(1):1–28
14. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics* 21(1):1–13
15. Ali H, Rada L, Badshah N (2018) Image segmentation for intensity inhomogeneity in presence of high noise. *IEEE Trans Image Process* 27(8):3729–3738
16. Fang J, Liu H, Zhang L, Liu J, Liu H (2021) Region-edge-based active contours driven by hybrid and local fuzzy region-based energy for image segmentation. *Inf Sci* 546:397–419
17. Alpert S, Galun M, Basri R, Brandt A (2007) Image segmentation by probabilistic bottom-up aggregation and cue integration. In: 2007 IEEE conference on computer vision and pattern recognition, pp 1–8
18. Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A (2020) Dataset of breast ultrasound images. *Data Brief* 28:104863

# An Improved Intrusion Detection System for the Internet of Medical Things Based on Deep Convolutional Neural Network



Pandit Byomakesha Dash, H. S. Behera, and Manas Ranjan Senapati

**Abstract** Internet of Things (IoT) developing technologies have opened a new chapter in healthcare. IoT technology has revolutionized the delivery of healthcare to patients. As network-enabled IoT devices are incorporated into the Internet of Medical Things (IoMT) network architecture, the safety of IoMT networks is a major concern for healthcare. This article provides a detection strategy based on deep learning (DL) for identifying malicious activity in the IoMT network settings. The suggested deep convolutional neural network (DCNN) detection approach determines the traffic's network protocol before detecting IoMT abnormalities. As the majority of data is imbalanced with a small percentage of spoofing and data injection assaults, this study demonstrates how unbalanced data potentially affect training process of the model and provides an effective strategy synthetic minority oversampling technique (SMOTE) to deal with such a scenario. The simulation findings demonstrate that the suggested DCNN classification model achieves superior results compared to state-of-the-art classifiers.

**Keywords** IoMT · Healthcare · Deep learning · Intrusion detection system · DCNN

---

P. B. Dash (✉) · H. S. Behera · M. R. Senapati

Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Sambalpur, Odisha 768018, India

e-mail: [byomakeshdash2000@gmail.com](mailto:byomakeshdash2000@gmail.com)

H. S. Behera

e-mail: [hsbehera\\_india@yahoo.com](mailto:hsbehera_india@yahoo.com)

M. R. Senapati

e-mail: [mrsenapati\\_it@vssut.ac.in](mailto:mrsenapati_it@vssut.ac.in)

## 1 Introduction

The growth of information and communication technology has made a significant impact on healthcare technologies in a wide range of fields during the last decade. IoT is a network-based technology that allows objects to share data and interact with their physical surroundings in various ways. IoT provides a vast array of applications, including intelligent homes, intelligent transportation, and intelligent healthcare [1]. The demand for IoT in healthcare has increased significantly in recent years. Some examples of medical IoT devices include pulse oximeters, blood glucose monitors, asthma inhalers, and other wearables also known as the IoMT devices used extensively by healthcare institutions in cyber-physical systems (CPS). IoT healthcare applications may monitor patients, specimens, and supplies, as well as enhance service quality and staff productivity, by making use of biometrics information and measures obtained by sensors [2].

The elimination of the need for routine checkups at the doctor's office is one of how the IoMT has contributed to improvements in the healthcare sector. Patients were able to gain insight into their health and the progression of their treatment, costs were reduced, healthcare providers were able to better assist their patients, and patients were able to prescribe more appropriate medications as a result of the improved ability of healthcare providers to communicate clearly and straightforwardly. Because these systems handle confidential and frequently life-threatening medical information, widespread deployment of the IoMT may be delayed as a result of security and privacy concerns. Patients might suffer serious bodily injury or even death if cybercriminals have been able to compromise the IoMT's most vital security features, including its privacy, safety, reliability, and authenticity. Insulin overdose is one conceivable outcome of hacking a medical insulin pump, for instance. The life of a patient might be in danger if a linked cardiac device is hacked.

Due to the unique characteristics of IoMT devices, typical security solutions are inadequate for properly detecting IoMT cyberattacks. Therefore, designing a security strategy for IoMT devices is quite complex. The majority of recent research has been on developing authentication and encryption-based solutions for IoMT devices. Additionally, the method of intrusion detection is made more complicated by the fact that IoMT devices may use a wide variety of communication protocols, including EtherCAT, Ethernet, EtherNet/IP, TCP/IP, and Modbus. To secure the healthcare industry from cyberattacks, an intrusion detection system (IDS) is an excellent defensive mechanism, particularly when it comes to identifying suspicious activities in the specified IoMT network setting.

Data analysis-based approaches have been employed in this study because they are more efficient and effective in dealing with the unanticipated complications that arise from cyberattacks of unknown origin. The framework's main goal is to construct an intelligent, safe, and trustworthy IoMT-based system that can detect and fix vulnerabilities, defend itself from cyberattacks, and then recover subsequently. In order to classify and identify patterns, DL utilizes a multi-tiered, hierarchical, data-processing architecture. In the modern century, DL has been successfully used in a broad variety

of applications as a result of its achievements and its stability. Some examples include natural language processing, computer vision, and cybersecurity systems.

The most important contributions of this research are summarized here:

- The implementation of a reliable, adaptable, and fault-tolerant framework for anomalies detection in the IoMT networks.
- This research develops and evaluates a deep convolution neural network (DCNN) for IoMT intrusion detection on a deep learning platform.
- This article combines DCNN with the SMOTE oversampling method to handle the issue of an unbalanced dataset.
- The effectiveness of the suggested approach is illustrated by comparing it to existing machine learning approaches.

This study is organized as follows: Sect. 2 will provide a summary of early studies, while Sect. 3 will provide an overview of the framework. Section 4 offers dataset discussion and environmental setup. Section 5 presents the research results, and Sect. 6 concludes up the research.

## 2 Literature Study

Some studies of IDSs that combine machine learning methods and classifiers to find intrusions in IoMT networks are shown here. Kaplantzis et al. [3] have suggested a centrally managed IDS using support vector machines (SVMs) and sliding windows. Without reducing the power of network nodes, the suggested approach can effectively identify black-hole attacks with a detection rate of 100% and selective forwarding intrusions with an accuracy rate of 85%. In [4], a distributed DL-based IoT/Fog network IDS has been designed to monitor the distributed architecture of IoT applications and identify any IoT assaults that may occur. Specifically, the NSL-KDD traffic distribution dataset has been used for this evaluation. As shown by experimental results, a DL-based distributed attack detection system outperforms a model trained with more conventional ML algorithms such as SVM, decision trees (DTs), and other neural networks in terms of a variety of performance metrics. Using DL, the suggested model's total accuracy rose from around 96% to over 99%. This study is more beneficial than the distributed design of IoT apps for accurately identifying IoT assaults.

A bi-directional long short-term memory neural network (B-LSTM) technique has been suggested for intrusion detection by Roy et al. [5], which attempts to distinguish between normal and attack behaviors using binary classification. The proposed framework is over 95% accurate in detecting IoT attacks after being trained on the UNSW-NB15 dataset. Almiani et al. [6] suggested the implementation of a deep recurrent neural network (RNN) model for IoT devices. DoS, Probe, U2R, and R2L attack detection rates were determined to be 98.27%, 97.35%, 64.93%, and 77.27%, respectively, utilizing the NSL-KDD dataset to test the performance. Using information extracted from the system call graph, Le et al. [7] established a CNN-based botnet

detection model with a single class classification. This model has an F-measure of 98.33% and an accuracy of 97%.

To classify a wide range of known intrusions, Li et al. [8] used GRU, LSTM, broad learning system (BLS), and BiLSTM techniques on the NSL-KDD. According to the results of the performance evaluation, the BLS speeds up the model training process while maintaining an accuracy of 84.15% on the KDDTest + dataset and 72.64% on the KDDTest-21 dataset. The RNN-IDS accuracy ranged from 85.5% to 95.25% in a study conducted by Ayyaz et al. [9]. Initially, the IDS is trained using a gradient descent approach, then it is trained and evaluated using the KDD20+ and KDDTest+ datasets. RNN-IDS outperforms the other used algorithms, including, SVM, J48, NB, MLP, RF, and ANN, in terms of accuracy and speed.

Jiang et al. [10] have proposed a hybrid sampling-based anomaly detection approach, which they investigated on both the NSL-KDD and UNSW-15 datasets. Models constructed using CNN-BiLSTM, CNN, RF, BiLSTM, LeNet-5, and AlexNet classifiers are trained on a balanced dataset comprised of SMOTE and OSS. Statistics show that CNN-BiLSTM outperforms competing algorithms, with accuracy 83.58% and 77.16%, respectively, on the test datasets. Another ML-based intrusion detection strategy has been proposed by Sahu et al. [11], which uses a concurrent implementation of LR and ANN classification techniques. When examining the entire dataset, LR and ANN both have around 99.4% accuracy, whereas improving to 99.99% accuracy requires eliminating approximately 105,952 records from the original data. The original dataset is separated into a training set of 75% and a test set of 25%.

For healthcare and other IoT-based applications, Saif et al. [12] have created a hybrid intelligent intrusion detection system (HIIDS) based on machine learning and metaheuristic algorithms. The proposed HIIDS's performance has been measured against that of the widely used NSL-kDD dataset, which consists of 41 features and 125,973 samples. To reduce computing cost, supervised learning algorithms like kNN, DT, and a population-based metaheuristic algorithm like genetic algorithm (GA) have been designed for efficient categorization of anomaly and normal type based on chosen features, while metaheuristic algorithms like differential evaluation (DE), GA, and PSO are adopted for appropriate feature selection. The GA-DT approach has obtained a highest attack performance rate of 95.71%.

To handle growing cyber-risks in the domains of electronic health records (EHRs) and Internet-connected medical devices (IoMT's), Wahab et al. [13] have proposed an AI-driven, SDN-enabled IDS. A combination of a long short-term memory (LSTM) with gated recurrent unit (GRU) has been implemented. The CICDDoS2019 dataset, which is available to the public, was used to perform extensive testing on the suggested model. The proposed model had a 99.01% accuracy rate, a 99.04% precision rate, a 98.80% recall rate, and a 99.12% F1-score.

This literature review highlights the following limitations in research. IoMT environments in real-time contain massive amounts of unpredictable data. Training with unpredictable data bounds using a conventional machine learning model has an effect on accuracy performance. Several ML and DL models processing irrelevant features reduce the accuracy of IDS and extend training time. DL-based techniques are capable

of resolving all of the mentioned issues. DCNN is the most popular and well-known algorithm in the field of DL. The primary advantage of DCNN over its predecessors is that it automatically recognizes significant features without human intervention. The primary advantage of DCNN is the weight-sharing feature, which decreases the amount of trainable network parameters, hence enhancing generalization and preventing overfitting. DCNN makes the implementation of large-scale networks simpler than other neural networks.

### 3 Proposed Work

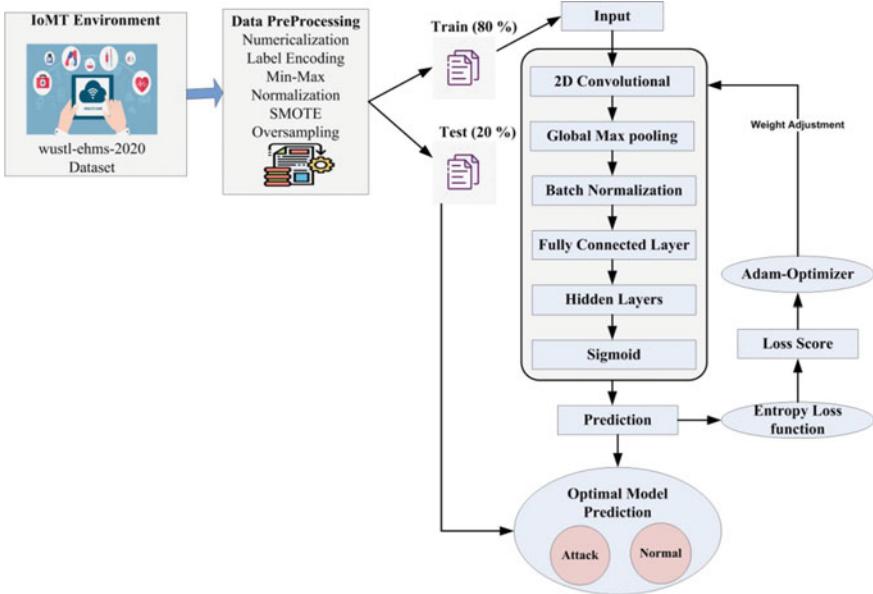
The goal of DL, a subfield of machine learning, is to progressively replicate high-level properties by using successive layers. DL has the potential to be quite domain-specific and adaptive ability. DL model comprises an input layer, set of hidden layers, and an output layer.

#### 3.1 *Proposed DCNN*

This section describes the implementation of our proposed classification algorithm for anomaly detection in the IoMT environment. Figure 1 illustrates the proposed scheme's workflow. This research utilized the WUSTL-EHMS-2020 dataset, which is a good IDS dataset that has been widely utilized in recent studies. This study has four phases. Phase 1 consisted of data collecting from an IoMT environment in real time. Phase 2 was the preliminary data processing. The missing and duplicate values were correspondingly replaced and eliminated. Outliers were also checked. Min-max normalization was used to reduce a dataset to a single standard scale. Encoding of data was also conducted. To address the unbalanced nature of the dataset, we have employed the SMOTE oversampling approach. The third phase consisted of separating the data into training (80%) and testing (20%) sets. In Phase 4, DCNN classifiers were deployed for IoMT network classification.

Specifically, we have suggested a DL-based deep convolution neural network (DCNN). The layers of convolutional neural networks may be categorized according to their functions. The suggested approach is based on the structure of DCNN, which is implemented by a sequence of steps known as the two-dimensional convolution layer, global pooling layer, batch normalization layer, and fully connected layer. In the suggested models, we have two convolution layers, two max-pooling layers, a regularization layer, a completely connected layer, and an output layer. Figure 1 demonstrates the proposed structure. The proposed DCNN model requires 216 s to execute after being trained using 50 epochs, batch size = 250, and a learning rate of 0.01.

The suggested model's first layer is the convolution layer. The first hidden layer contains 43 neurons. The chosen area of neurons is interconnected to a single neuron



**Fig. 1** Proposed DCNN workflow framework

that resides in second hidden layer. The second layer will have a unique neuron. Using the weight matrix and bias, the transition from one hidden layer to the next is determined. To illustrate the correlation between bias and weight, Eq. (1) is used.

$$f(X_i, W_i, b) = \sum W_i X_i + b \quad (1)$$

where:  $X_i$  represents an instance,  $W$  and  $b$  represent weight and bias, respectively.

To create an activation map, the filters are applied to a segment of the convolution layer to set its size. By using stride, the next layer's depth could be minimized. The desired overlap between two layers' output values can be computed using this approach. Table 1 presents the algorithm for the proposed DCNN.

With a filter scale of 2 and a stride value of 1, the first and second 2D generic convolution layers each use 64 filters. Until it is pooled, the model learns increasingly complex functions in the second layer. The filter scale is 2, while the stride value is 1 in the two-dimensional global maximum pooling layer. In this layer, the greatest value of the filtered data substitutes for it. Taking the maximum value prevents the analyzed functions from undergoing overfitting. The data from the previous layer is normalized in a batch by the normalization layer before it is used by the following layer.

The completely linked layer consists of neurons that have connections to all of the other neurons in both the current and the preceding layers. The trust findings are generated by a single neuron in the final, output layer using a cross-entropy-based

**Table 1** Algorithm based on DCNN

---

**Input:** E: Number of epochs; W: weight parameters of DCNN;  $\alpha$ : learning rate; b:batch sizes;  $X_{train}$ : training data;  $X_{test}$ : test data; X': WUSTL-EHMS-2020 dataset

---

**Output:**  $P_{test}$ : Performance metrics are computed from test data

---

Initialize all the corresponding layers (convolution layer, pooling layer, normalization layer, flatten layer, and hidden layers) of DCNN model with parameters  $\theta$

---

**Anomaly Classification Training:**

---

Preprocess the healthcare dataset

---

$X_{train} \leftarrow \text{PreData}(X_{train})$

---

$X_{test} \leftarrow \text{PreData}(X_{test})$

---

**While**  $\theta$  has not converged **do**

---

**For** local epoch 1 to E **do**

---

**For** s = (x,y)  $\in$  random batch from  $X_{train}$  **do**

---

            Update model parameters  $\theta$

---

**End**

---

**End**

---

**End**

---

Evaluation of trained DCNN model with test samples to compute  $P_{test}$

---

sigmoid activation. The result of a completely linked layer using the sigmoid function is described by Eq. (2).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

Equation (3) defines the entropy cross-correlation between the measured output and the measured label:

$$h(P, Q) = \sum P(X) \log Q(X) \quad (3)$$

## 4 Dataset Analysis and Simulation Setup

This section is describing all details of dataset instances and work carried out in hardware and software simulation environments.

#### 4.1 Overview of Dataset

Using a real-time enhanced healthcare monitoring system (EHMS) testbed, the WUSTL-EHMS-2020 dataset was generated [1]. Due to the lack of a dataset that integrates various biometrics, this testbed gathers both network traffic measurements and biometrics from patients. Man-in-the-middle attacks like as spoofing and data injection are represented in this dataset. The spoofing attack's primary function is to intercept on traffic between the gateway and the target server, violating the patient's right to privacy. The data injection attack modifies packets on-the-fly, hence violating the data's integrity. The 44 features in this dataset include 35 network flow measurements, eight patient biometric variables, and one label feature. There are a total of 16,318 records in the dataset. There have been 14,272 healthy samples (87.5%) and 2046 malicious samples (12.5%).

#### 4.2 Simulation Environment

In this work, machine learning models are written in Python and run in the conda environment. Python 3.8.8 is used by the 4.10.1 conda version. Python modules such as scikit-learn, pandas, NumPy, and Matplotlib are being utilized to put this work into reality. In this work, we utilized a computer with an Intel®CoreTM i7-8550U CPU as our primary hardware platform (CPU). The primary clock speed is 1.80 GHz, and there is 8 GB of RAM on board (RAM). Intel®UHD Graphics 620 is the system's GPU. Other than that, Windows 10 Home 64-bit is running on the computer.

### 5 Result Analysis

In this study, we compared the results of many popular machine learning algorithms, dissecting their performance using a variety of different measures and explaining the results using DCNN. Results are shown using several measures (precision, recall, F1-score, and ROC-AUC, and Accuracy), with the acquired accuracy serving as an approximation for the complete performance of the suggested approach. In this study, the distribution of class labels is very asymmetric and non-uniform; hence, F1-score is a useful metric for precisely measuring performance. In our analysis, we used a broad array of examples, each with its unique features. With this objective, we conduct experiments with optimizers' learning rates, dropout percentages, activation functions for hidden layers, and more. There is a significant impact on the calculated performance metrics of the DL-based DCNN model from these essential parameters.

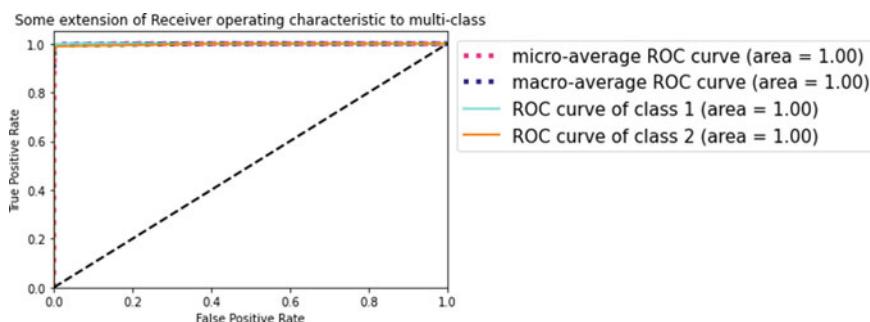
Table 2 demonstrates a comparison of advanced ensemble ML techniques, including Random Forest (RF), Bagging, AdaBoost, Gradient Boosting (GBoost), Extreme Gradient Boosting (XGBoost), CatBoost, and the suggested DL-based

DCNN. When compared to more advanced ensemble ML algorithms, CatBoost has the greatest F1-score (0.9953), but Bagging has a distinctly average F1-score (0.9678). The suggested DCNN model has an accuracy score 99.92%, a precision score 0.9984, a recall score 0.9986, an F1-score 0.9988, and a ROC\_AUC score 0.9999.

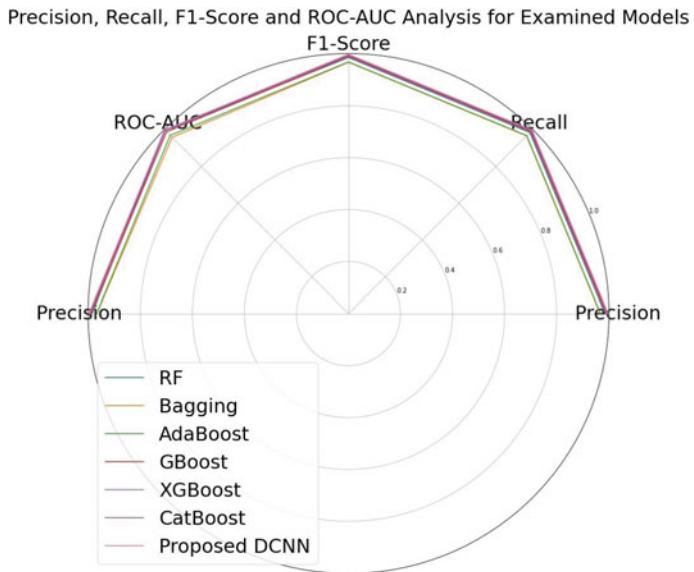
Figure 2 depicts the AUC-ROC study for the DL-based DCNN approach. It has been shown that the DCNN methodology is superior to conventional comparison approaches when all classes' coverage areas have been taken into consideration. Figure 3 shows a comparison between the performance of the specified advanced ensemble ML model and the proposed DCNN Method. The suggested DCNN model achieves superior accuracy of 99.92%, precision of 0.9984, recall of 0.9986, F1-score of 0.9988, and ROC-AUC score of 0.9999. In Fig. 4, the accuracy measure is used to compare the evaluated advanced ensemble machine learning and proposed DCNN classifier. Figure 5 depicts the class-wise classification analysis for suggested model.

**Table 2** Proposed model compared with other ML models

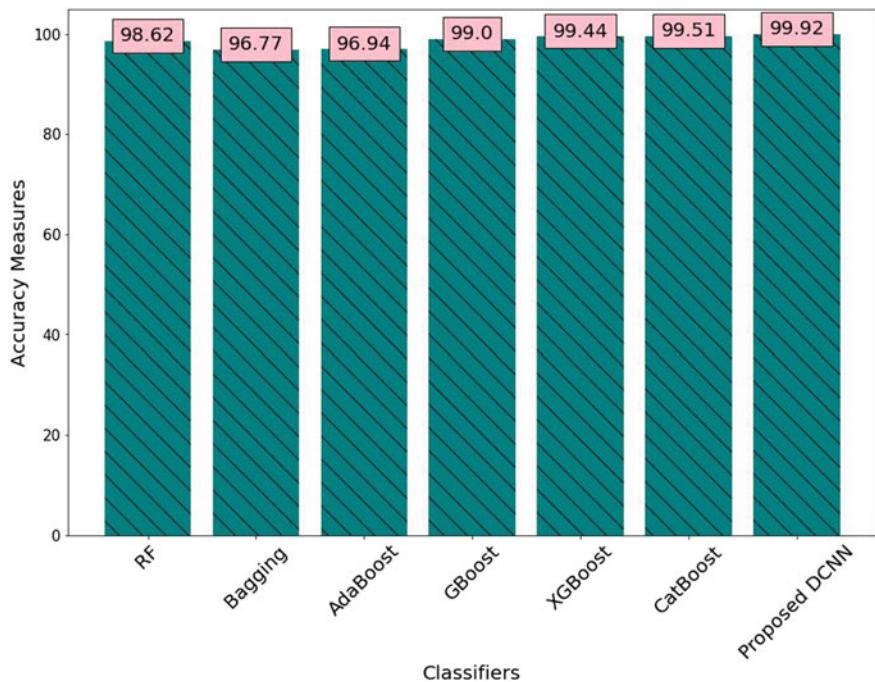
Prediction models	Performance measures				
	Precision	Recall	F1-score	ROC-AUC	Accuracy (%)
RF	0.9861	0.9857	0.9862	0.9955	98.62
Bagging	0.9675	0.9674	0.9678	0.9594	96.77
AdaBoost	0.9689	0.9688	0.9692	0.9697	96.94
GBoost	0.9898	0.9899	0.9900	0.9892	99.00
XGBoost	0.9935	0.9932	0.9931	0.9961	99.44
CatBoost	0.9951	0.9950	0.9953	0.9985	99.51
Proposed DCNN	0.9984	0.9986	0.9988	0.9999	99.92



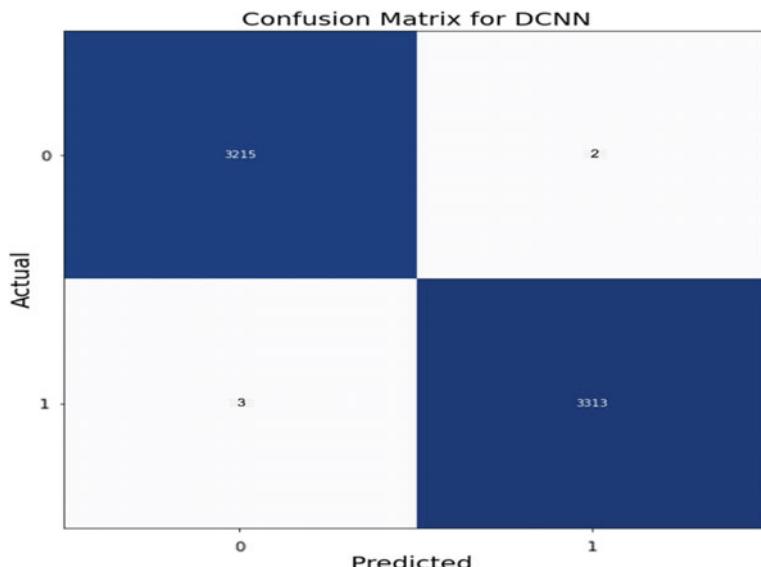
**Fig. 2** ROC-AUC analysis of suggested DCNN



**Fig. 3** Performance analysis for implemented models



**Fig. 4** Accuracy comparisons for examined models



**Fig. 5** Confusion matrix analysis for proposed model

## 6 Conclusion

The suggested model has been implemented using DCNN since it is a more adaptable algorithm with a significant level of accuracy. The suggested DCNN-based model can identify IoMT spoofing and data injection attacks and has recognized suspicious network traffic in real time. Using the suggested DCNN-based model, the experimental findings demonstrated accurate IoMT attack detection and classification. In addition, many advanced ensemble machine learning algorithms, including RF, Bagging, AdaBoost, GBoost, XGBoost, CatBoost, and KNN, were evaluated to examine the performance of the suggested approach. The method provided by DCNN achieves values that allow for very accurate attack detection. The proposed DCNN achieves accuracy score 99.92%, precision score 0.9984, recall score 0.9986, F1-score 0.9988, and ROC-AUC score 0.9999. Future research will be conducted on a dataset with a larger number of multiclass cases.

## References

1. Kumar P, Gupta GP, Tripathi R (2021) An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for iomt networks. *Comput Commun* 166:110–124
2. Swarna Priya RM, Maddikunta PKR, Parimala M, Koppu S, Gadekallu TR, Chowdhary CL, Alazab M (2020) An effective feature engineering for dnn using hybrid pca-gwo for intrusion detection in iomt architecture. *Comput Commun* 160:139–149

3. Kaplantzis S, Shilton A, Mani N, Ahmet Sekercioglu Y, Detecting selective forwarding attacks in wireless sensor networks using support vector machines. In: 2007 3rd International conference on intelligent sensors, sensor networks and information, 3–6 Dec 2007, Melbourne, Qld., Australia, pp 335–340. <https://doi.org/10.1109/ISSNIP.2007.4496866>
4. Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for Internet of Things. *Fut Gener Comput Syst* 82:761–768. <https://doi.org/10.1016/j.future.2017.08.043>
5. Roy B, Cheung H (2018) A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In: 2018, 28th International telecommunication networks and applications conference, Sydney, NSW, Australia, pp 1–6
6. Almiani M, AbuGhazleh A, Al-Rahayfeh A, Atiewi S, Razaque A (2020) Deep recurrent neural network for IoT intrusion detection system. *Simul Model Pract Theory* 101:102031–102056
7. Le H-V, Ngo Q-D, Le V-H (2019) IoT botnet detection using system call graphs and one-class CNN classification. *Int J Innov Technol Exploring Eng* 8(10):937–942
8. Li Z, Batta P, Trajkovic L (2018) Comparison of machine learning algorithms for detection of network intrusions. In: 2018 IEEE International conference on systems, man, and cybernetics, Miyazaki, Japan, pp 4248–4253
9. Ayyaz-ul-Haq Qureshi B, Larjani H, Ahmad J, Mtetwa N (2019) A heuristic intrusion detection system for internet-of-things (IoT). In: Intelligent computing: proceedings of the 2019 computing conference, London, United Kingdom, pp 86–98
10. Jiang K, Wang W, Wang A, Wu H (2020) Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access* 8:32464–32476
11. Sahu NK, Mukherjee I (2020) Machine learning based anomaly detection for IoT network: (anomaly detection in IoT network). In: 2020 4th International conference on trends in electronics and informatics (48184), Tirunelveli, India, pp 787–794
12. Saif S et al (2022) HIIDS: hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IoT based healthcare. *Microprocess Microsyst* 104622. <https://doi.org/10.1016/j.micpro.2022.104622>
13. Wahab F et al (2022) An AI-driven hybrid framework for intrusion detection in IoT-enabled E-health. *Comput Intell Neurosci* 2022:6096289. <https://doi.org/10.1155/2022/6096289>

# Deep Convolutional Neural Network for Skin Cancer Classification



Pandit Byomakesha Dash, Ch Ravi Kishore, Venkatasai Kommu,  
Vysyaraju Lokesh Raju, and Subhasree Mohapatra

**Abstract** The most prevalent kind of cancer that poses a significant threat to human health and survival is skin cancer. Convolutional neural networks (CNNs) in deep learning (DL) have shown promising results in reliably detecting skin cancer, which might be useful for the job of imaging experts. To achieve an effective skin cancer diagnosis, training a CNN model on a bigger dataset of labeled skin cancer images is suggested. However, additional labeling for skin cancer images is not always accessible. The main objective of this research is to use a CNN with hidden layers for accurate skin cancer classification. We have presented two unique classification models, including CNN with a single hidden layer and CNN with two hidden layers. Here, the CNN model for skin cancer categorization relies heavily on its hidden layers. The single-layer CNN model gives 100% accuracy for skin cancer diagnosis in 32.64 s, while the dual-layer CNN model takes 38.45 s to complete its task. The purpose of this study is to find a way to make preprocessed skin cancer images work with CNN by adding additional hidden layers to the network.

---

P. B. Dash (✉)

Department of Information Technology, Aditya Institute of Technology and Management, Tekkali, Kotturu, Srikakulam, Andhra Pradesh 532201, India  
e-mail: [byomakeshdash2000@gmail.com](mailto:byomakeshdash2000@gmail.com)

C. R. Kishore · V. Kommu

Department of Computer Science and Engineering, Aditya Institute of Technology and Management, Tekkali, Kotturu, Srikakulam, Andhra Pradesh 532201, India  
e-mail: [cauchy9@gmail.com](mailto:cauchy9@gmail.com)

V. Kommu

e-mail: [venkatasai7425@gmail.com](mailto:venkatasai7425@gmail.com)

S. Mohapatra

Institute of Technical Education and Research, Sikhsa ‘O’ Anusandhan University, Bhubaneswar, Odisha 751030, India  
e-mail: [subhashreemohapatra@soa.ac.in](mailto:subhashreemohapatra@soa.ac.in)

V. L. Raju

Tata Consultancy Services Pvt Ltd, Pune, Maharashtra, India  
e-mail: [lokeshrajuvysyaraju1@gmail.com](mailto:lokeshrajuvysyaraju1@gmail.com)

**Keywords** Skin cancer · CNN · Deep learning · Hidden layer · Artificial intelligence

## 1 Introduction

There has been a rise in the number of people diagnosed with cancer in recent years, and researchers believe that this is related to changes in people's diets, levels of physical activity, exposure to sunlight and other forms of radiation, infections, and other causes. Skin cancer is the most frequent form of the disease. According to WHO estimations, the number of persons with skin cancer would increase to about 13.1 million by the year 2030 [1, 2]. Cancer of the skin may manifest as the abnormal growth of skin cells. Actinic keratoses (AK), basal cell carcinoma (BCC), squamous cell carcinoma (SCC), and melanoma are the four primary types of skin cancer. It affects both sexes and accounts for roughly 75% of skin cancer-related mortality [3]. According to the world cancer report, ultraviolet light exposure is the primary cause of melanoma in those with poor skin pigmentation. Approximately, 25% of malignant moles [4] are attributed to exposure to ultraviolet light, which may occur from the sun or other sources.

Utilizing a variety of techniques, physicians can identify skin cancer. Visual identification is the first step in determining the disease's potential [5, 6]. ABCD (asymmetry, border, color, diameter) is a guide devised by the American Center for the Study of Dermatology [7] for determining the likely form of melanoma. It is used by physicians for early disease screening. If a suspicious skin lesion is discovered, the physician obtains a skin sample and examines it under a microscope to determine whether the lesion is malignant or benign. Dermatoscopy is a procedure used to diagnose skin cancer [8]. It entails photographing the contour of the skin lesion, which appears as black dots, in high-contrast light [9]. However, this approach has some limitations, the most prominent of which is the difficulty to detect the type of the lesion owing to hair, blood vessels, brightness, the failure to capture the proper form of the spot, and the similarity in appearance between malignant and noncancerous areas.

DL has transformed the whole landscape of machine learning (ML) in recent decades. It is the area of ML that deals with the algorithms of artificial neural networks and is considered the most advanced of the subfields. The operation and makeup of the human brain served as a model for the development of these algorithms. There is a wide variety of applications for DL methods, including voice recognition, pattern recognition, and bioinformatics. DL systems have been demonstrated to produce outstanding results in a variety of applications, particularly when contrasted with more traditional techniques of ML. Several deep-learning methods have been used in recent years for the computer-based identification of skin cancer. Early detection is essential for treating skin cancer, which is a growing health risk. It's not only expensive, but the diagnosis is also labor-intensive and time-consuming. However,

modern science having advanced ML and DL may be useful in several ways. Consequently, ML and DL may facilitate the detection of diseased cells, and DL-based neural networks, in particular convolutional neural networks, are utilized to identify cancerous cells more rapidly and effectively.

This study includes the following contributions:

- Create a dermoscopy image-based skin cancer screening model that uses artificial intelligence to detect melanoma and non-melanoma. This AI-based model has the potential to enhance clinical screening tests, decrease diagnostic mistakes, and promote earlier disease detection.
- To distinguish between malignant and noncancerous skin lesions, a Deep Convolutional Neural Networks (DCNN) model has been designed.
- Higher importance has been assigned to F1-Score, Precision, Recall, ROC-AUC, and Accuracy as standard metrics for evaluating classification performance.
- This study's main contribution is proof that hidden layers are essential to the DCNN design process. Since only one hidden layer is used in DCNN creation, the model's complexity is kept to a minimum while still providing optimal accuracy over many runs, while DCNN's two hidden layers produce the best accuracy at the earliest epoch, at the expense of a longer computation time.

The remainder of this research is structured as follows: literature work presented in Sects. 2, and 3 outlines DL methods in depth, Sect. 4 provides an overview of the dataset and simulation environment, Sect. 5 presents performance evaluation results and discusses the models, and Sect. 6 concludes.

## 2 Literature Survey

Multiple studies have detected, segmented, and classified skin tumors using computer vision, ML, image processing, neural network, and classification methods. Sultana et al. [10] have suggested an alternative DL-based method for the early identification prediction of melanoma from skin lesions. Multi-layer neural networks receive input in vector form, while convolutional networks may process both organized and unstructured data. An accuracy of 83.09% was achieved by using the ISIC dataset of image categorization. To detect skin lesions in the previse stage, Haenssle et al. [11] used a Google Net Inception CNN architecture that has already been trained, in addition to 100,000 digital pictures. The research employed the Deep CNN model as a reference point for comparison to the CNN, which had a mean sensitivity of 86.6% for diagnostic categorization. Achim Hekler et al. [12] have suggested a DL model for early diagnosis of melanoma of the skin. This study combines the knowledge of dermatologists with DL and AI to create the primary digital skin diagnostics approach. They combine the findings of dermatologists and CNN using ResNet50 and XGBoost models. By including human judgment, they were able to examine 100,000 images from the HAM10000 Dataset and find that the sensitivity rose from

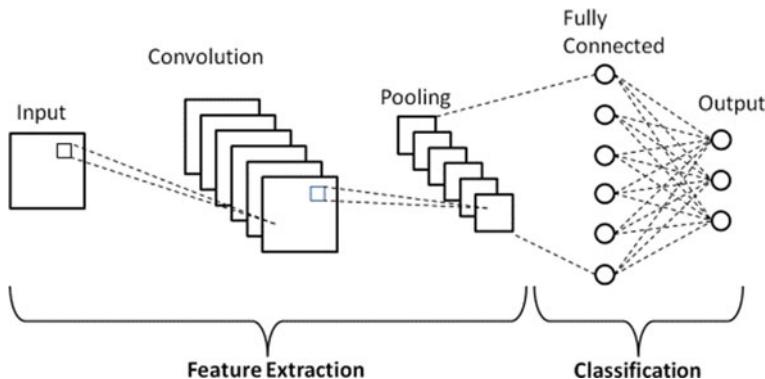
86.1% to 89%, which was statistically significant ( $p < 0.05$ ). The CNN was able to attain a sensitivity of 89% but only a specificity of 81.5%.

Fuadah et al. [13] have created an effective model for predicting early-stage skin cancer. They analyzed a freshly produced picture using the Net, Atlas, and ISIC dataset classification methods. Alex Net, VGG16, and ResNet-18 pre-trained DL models analyzed 150 genuine ISIC test images. From this detected accuracy of 93.29%, their model has achieved an accuracy of 83.83% in the categorization of melanoma and 97.55% in the classification of seborrheic keratosis. Using CNN-based histologic pictures, Hohn et al. [14] proposed that early acquisition of data that improves the prediction of skin cancer is potential. Highly accurate results were obtained through CNN-based image analysis, and patient data integration was used to identify a small number of lesions that were not included in CNN's high-confidence classifications. The convolutional neural network model was applied to the full-size picture (WSL) dataset with an accuracy of 92.30%.

Patil et al. [15] have proposed ML models for the classification of malignant melanoma. There were three distinct algorithms, including CNN, SVM, and XGB. This study employs the ISIC dataset, which consists of 2000 images. The CNN + SVM achieved 84.0% accuracy and CNN + XGB achieved 88.0% accuracy predicted for melanoma skin cancer. Sujaini et al. [16] suggested a linear deep-learning method for identifying melanoma skin cancer. The two ML and DL models Type1 and Type2 utilized in the ISIC dataset to predict skin cancer are linear regression and CNN. CNN is superior to linear regression for the detection of melanoma. For the prediction of melanoma skin lesions, linear regression reaches an accuracy of 68% and a sensitivity of 56%, while CNN approaches an accuracy of 70% and a sensitivity of 75%. AlShourbaji et al. [17] suggested a method for the early diagnosis of skin lesions. The results show that DL algorithms, particularly RNN, can accurately identify melanoma and benign cases from X-ray pictures with an accuracy of 82.4%. The HAM10000 dataset contains both test data and training data images of melanoma and benign cases.

### 3 Methodology

CNN is an enhanced form of the neural network. CNN is composed of neurons whose structure resembles that of human neuron cells. Each neuron in CNN gets input and has weights and biases that may be learned. In a typical artificial neural network, a single input vector is received, which is then processed and converted into a series of hidden layers. The disadvantage of a normal neural network is that it does not scale well to entire pictures, resulting in a lower classification accuracy than a CNN design, where the number of weights would expand dramatically in the event of huge images (length  $\times$  width  $\times$  color channels). Regular neural networks result in the development of a high number of parameters, which leads to the overfitting issue. CNN is specifically built for picture dataset categorization. Figure 1 depicts the architecture of CNN.



**Fig. 1** Architecture of CNN

CNN is supplied with preprocessed data. A CNN contains three types of layers. They are as follows.

- Convolutional Layer
- Pooling Layer
- Fully Connected Layer.

**Convolutional Layer:** The result of convolving an input of  $X * X$  with a  $Y * Y$  filter is represented in Eq. (1).

$$(X - Y + 1) * (X - Y + 1) \quad (1)$$

Image size reduction is a significant drawback of the convolution operation. When compared to pixels in the image's center, those at the image's corners are used much less frequently to compensate for data loss. To do this, an additional border was added to the image, padding it by one pixel in all directions. The padding approach has been expressed in Eq. (2), where  $P$  is denoting the padding value.

$$(X + 2P - Y + 1) * (X + 2P - Y + 1) \quad (2)$$

CNN's stride is an essential and beneficial element for reducing image size. For instance, when convoluting an image with a stride of 2, the horizontal and vertical directions are performed independently. The formula for stride  $s$  dimensions is expressed in Eq. (3), where  $Z$  denotes stride value.

$$[(X + 2P - Y)/Z + 1] * [(X + 2P - Y)/Z + 1] \quad (3)$$

Introducing the bias, the new function can be expressed in Eq. (4). Then, this equation transfer to the activation function Relu adopted here. Relu activation is expressed in Eq. (5).

$$Zi+ = bi + Xi * Wi \quad (4)$$

$$\text{Relu}(Zi) = \text{MAX}(0, Zi) \quad (5)$$

**Pooling Layer:** Pooling layers are commonly used to decrease image size and boost the computation speed.

The input of the pooling layer is  $Xh * Xw * Xc$ , the output can be presented in Eq. (6).

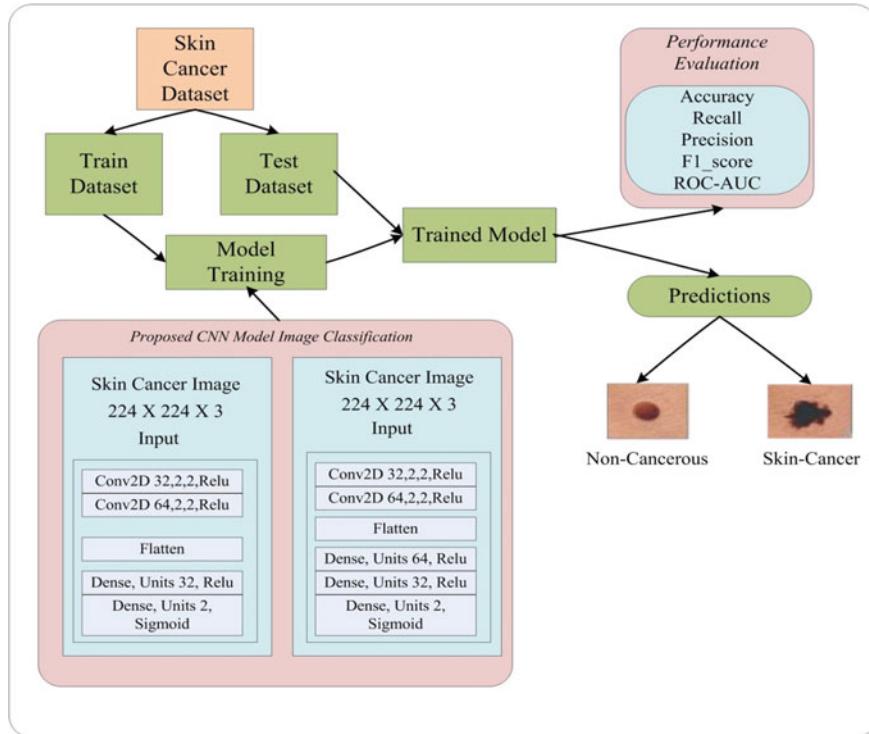
$$[(Xh - Y)/Z + 1] * [Xw - Y]/Z + 1 * Xc]. \quad (6)$$

Afterward, we employ convolution operation and pooling to extract increasingly complex features. The features are simplified to a single layer which is transfer to neural network with complete interconnection. By applying sigmoid as described in Eq. 7, the required benign or malignant outcome is produced.

$$\text{Output} = \frac{1}{1 + e^{-x}} \quad (7)$$

### 3.1 Proposed Methodology

Following scaling and normalization, the preprocessed images are utilized to train the CNN model. The standard picture size for input is  $224 \times 224 \times 3$  pixels. The suggested CNN model's convolution layer, the model's second layer, is responsible for choosing relevant features from images. Multiple  $3 \times 3$  pixel filters are used in the convolutional layer. Dot products are calculated on the pixel values when the filter is moved over the pictures in the convolution process. In this way, the results of the convolution operations, known as the convolved features, are generated. Dimensionality reduction is accomplished by performing a pooling operation on the acquired convolved features. The convolution layer is the most effective part of a CNN. Because it learns visual properties from very tiny squares of input data, this layer guarantees that neighboring pixels are physically adjacent. This is achieved by using a filtration system. All filters are  $3 \times 3$  in size. The output of such filters is calculated by taking the dot product of the input images and each filter's computations. We have chosen max pooling in our proposed study. Each layer, the convolutional one and the pooling one, uses the Relu function. For classification, this accumulated collection of features is used. In this case, we use a fully connected layer in which the nodes in the layer below the one being output have direct connections to each other. The class label has been determined using a sigmoid activation function in the output layer.



**Fig. 2** Architecture of proposed method

In this study, we utilize the skin cancer dataset to test two distinct CNN model architectures trained using the Adam optimization method on the task of identifying skin cancer types. Our first design is a CNN with a single hidden layer, and our second is a CNN with two hidden layers. The whole process for this proposed work is shown in Fig. 2. Table 1 represents the Hyperparameter setup for the proposed model.

#### 4 Experimental Setup and Dataset Overview

Experiments have been conducted using a Dell laptop equipped with Windows 11 Home Single Language 64-bit, an INTEL i7 Core processor, and 16 GB of RAM. Using Spyder IDE, the suggested models were constructed and evaluated in a Python Tensorflow environment. OpenCV is a part of the image processing programming environment that takes unprocessed images and turns them into a Numpy array so that they can be interpreted by a computer. Tensorflow, Keras, Numpy, Matplotlib,

**Table 1** Hyperparameters setup for the proposed model

Prediction models	Hyperparameters
CNN with single hidden layer	2D-Convolutional layer (no. of filters—64, filter size-2 *2, stride-1, activation function—Relu) No. of pooling layer-2 No. of batch normalization layer-1 Learning rate in Adam Optimizer-0.01 No. of hidden layer in FCN—1 No. of neurons in hidden layer—32 No. of epochs—50 Batch size—150 Output layer activation function—Sigmoid
CNN with two hidden layers	2D-convolutional layer (no. of filters-64, filter size-2 * 2, stride-1, activation function—Relu) No. of pooling layer-2 No. of batch normalization layer—1 Learning rate in Adam Optimizer—0.01 No. of hidden layer in FCN—2 No. of neurons in hidden layer—32 and 64 No. of epochs—50 Batch size—150 Output layer activation function—Sigmoid

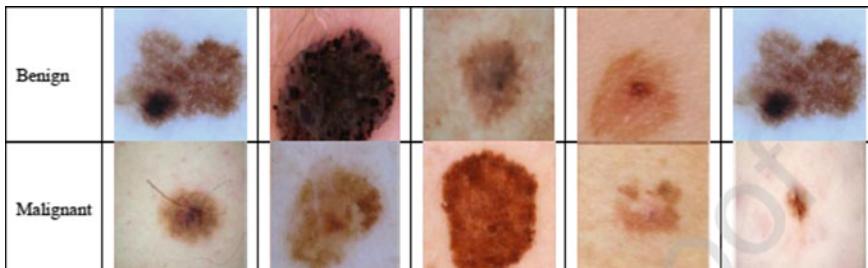
OS, Time, Random, and Pillow are a few of the other crucial packages used in this experiment.

#### 4.1 Overview of Dataset

The skin cancer Malignant versus Benign Dataset from Kaggle has been evaluated in our study. This collection includes images of both benign and cancerous moles. The data consist of two files, each containing 1800 images ( $224 \times 244$ ) of the two-mole species. About 80% of images of skin cancer Malignant versus Benign Dataset used for the training set, while the rest 20% is used for testing. Figure 3 represents some samples of benign and malignant types from the used dataset.

### 5 Analysis of Simulation Results

This skin cancer dataset has been used to test the suggested CNN architecture with binary class labels (malignant and benign tumors). By adding a denser layer to the Fully Connected CNN network, the suggested models' performances have been enhanced. To demonstrate the better performance of our two proposed architectures and to minimize the time gap between them, we have conducted comparison research



**Fig. 3** Benign and malignant samples from the original dataset

using CNN's two densest layers. In this research, the accuracy Eq. (8), sensitivity Eq. (9), and specificity Eq. (10) have been determined using the following formulas.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (8)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (9)$$

$$\text{Specificity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (10)$$

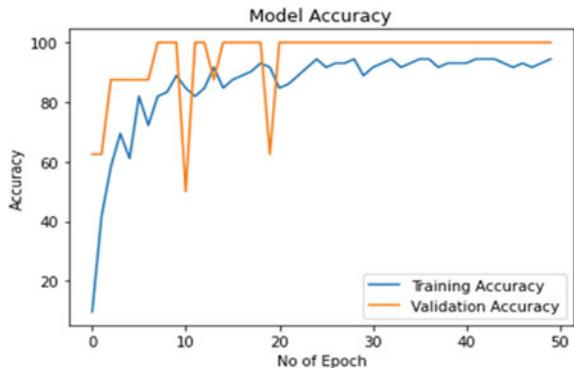
where TP indicates correctly identified healthy skin cases, TN indicates correctly categorized skin cancer cases, FP indicates incorrectly classified healthy skin cases, and FN indicates incorrectly identified skin cancer cases. In Table 2, we show a comparison of the performance metrics and execution time analysis of our suggested models. In this case, it is clear that both recommended models function well, but the model becomes better with fewer epochs and provides the greatest performance as the dense layer increases.

Our proposed CNN model accuracy is shown schematically in Figs. 4 and 5. Training accuracy for the proposed frameworks is shown to increase steadily from epoch 1 to epoch 35, peaking at 94% accuracy. Here, the two models exhibit distinct accuracy curves during testing. For the suggested CNN with one hidden layer, the test accuracy is 100% continuous starting at epoch 18 with no fluctuations in execution

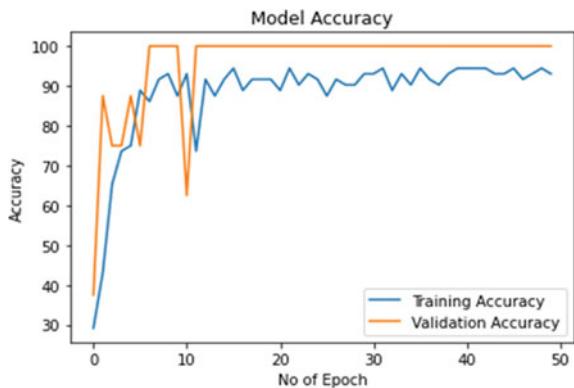
**Table 2** Performance comparison of proposed frameworks

Prediction Models	Precision	Recall	Accuracy	Execution time (s)
CNN with single layer (32 neurons)	1.0	1.0	99.96	32.64
CNN with two layers (32, 64 neurons)	1.0	1.0	99.99	38.45
Proposed model	1.0	1.0	99.99	38.45

**Fig. 4** CNN (one hidden layer)



**Fig. 5** CNN (two hidden layers)

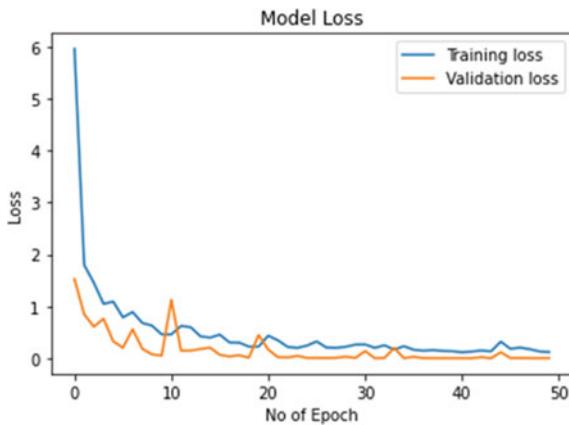


time, whereas, for the proposed CNN with two hidden layers, the test accuracy is 100% continuous starting at epoch 12 with no fluctuations in execution time.

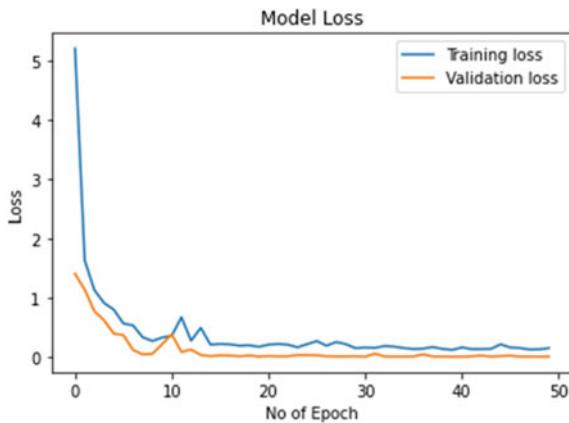
The loss curves for the two proposed CNN models are shown in Figs. 6 and 7, respectively. After 33 epochs, it has been found that the training loss for a CNN with a single hidden layer stabilizes around 0.27. In contrast, after 14 epochs of training, the loss in accuracy is consistently 0.14 for a CNN model with two hidden layers. By epoch 15, CNNs with two hidden layers are getting test losses of 0.05, while those with a single hidden layer are getting losses of 0.11.

Figures 8 and 9 show the results of a ROC analysis performed on the CNN classification models with one and two hidden layers, respectively. According to the findings, a CNN with two hidden layers is the most effective classifier for differentiating between various forms of skin cancer.

A few implementations of the proposed DCNN with two hidden layers are displayed in Fig. 10. As can be shown, the suggested model for skin cancer diagnosis is capable of not only segmenting the lesions in the standard databases but also detecting the lesion in images containing disturbances like body hair. The proposed



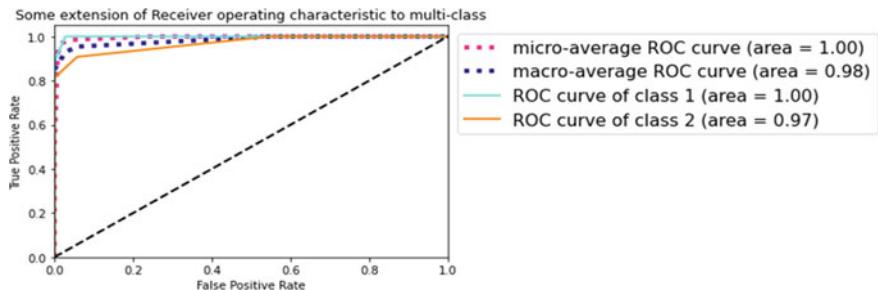
**Fig. 6** CNN (one hidden layer)



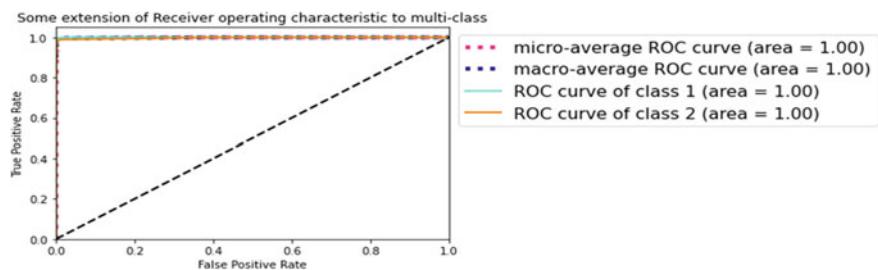
**Fig. 7** CNN (two hidden layers)

method's performance was found to be adequate for the diagnosis of skin cancer in different simulated environments.

Data from previous studies using the same experimental dataset are shown in Table 3. Previous studies show that the effectiveness of various classification measures varies from 89% to 99.20%, while the results obtained using our proposed method fall within the 99–1% range. As a result, it can be concluded that the proposed methodology performed better than the competing DL-based methods.



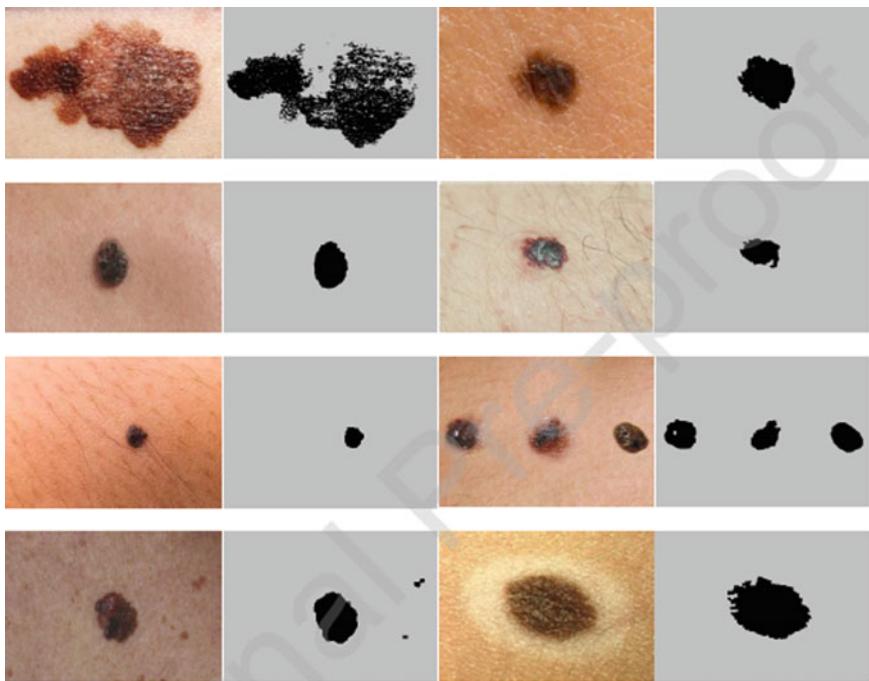
**Fig. 8** CNN (one hidden layer)



**Fig. 9** CNN (two hidden layers)

## 6 Conclusions

DL models have huge potential to increase the skin cancer screening detection rate. Our strategy may contribute to the future creation of a better system, such as CAD, that will assist the radiologist in identifying the most dangerous cases with the highest priority. Using CNN, we developed two classification models. CNN with a single hidden layer classification model produced a result of 100% accuracy with an execution time of 32.64 s, whereas CNN with two hidden layers classification produced the same result with an execution time of 38.45 s. It demonstrates conclusively that the hidden layer is a crucial component of CNN design for improved classification. In addition, we want to create a classification model for skin cancer with a greater number of class categories. Our detailed framework may be implemented in a different medical imaging dataset.



**Fig. 10** Some illustrations of using the proposed two hidden layer DCNN with input images and detected masks

**Table 3** Comparison of existing work with proposed model

Dataset used	Model implemented	Accuracy (%)	References
Malignant versus Benign Dataset	VGG-16	93.18	[18]
Malignant versus Benign Dataset	Stacking-CV	90.90	[19]
Malignant versus Benign Dataset	SkinNet-16	99.19	[20]
Malignant versus Benign Dataset	ResNet-152	89.65	[21]
Malignant versus Benign Dataset	DCNN with 2 hidden layers	100	Our work

## References

1. Cancer Research UK (2012) Cancer World Wide—the global picture. Retrieved Jan 16 2023 from <http://www.cancerresearchuk.org/cancer-info/cancerstats/world/the-global-picture/>
2. World Health Organization (2019) Skin cancer. Retrieved Jan 16 2023 from <http://www.who.int/en/>
3. Spencer Shawna Bram Hannah J, Frauendorfer M, Hartos Jessica L (2017) Does the prevalence of skin cancer differ by Metropolitan Status for males and females in the United States? *J Prev Med* 3, 3(9):1–6. <https://doi.org/10.21767/2572-5483.100019>

4. McGuire S (2016) World cancer report 2014. World Health Organization, International Agency for Research on Cancer, WHO Press, , Geneva, Switzerland. Advances in nutrition (Bethesda, Md.) vol. 7, 2 418–9. 15 2016. <https://doi.org/10.3945/an.116.012211>
5. Amin J, Sharif A, Gul N, Anjum MA, Nisar MW, Azam F, Bukhari SAC (2020) Integrated design of deep features fusion for localization and classification of skin cancer. *Pattern Recognit Lett* 131:63–70
6. Pathan S, Prabhu KG, Siddalingaswamy P (2018) Techniques and algorithms for computer aided diagnosis of pigmented skin lesions—a review. *Biomed Signal Process Control* 39:237–262
7. Chatterjee S, Dey D, Munshi S, Gorai S (2019) Extraction of features from cross correlation in space and frequency domains for classification of skin lesions. *Biomed Signal Process Control* 53:101581
8. Celebi ME, Kingravi HA, Uddin B, Iyatomi H, Aslandogan YA, Stoecker WV, Moss RH (2007) A methodological approach to the classification of dermoscopy images. *Comput Med Imaging Graph* 31:362–373
9. Goel N, Yadav A, Singh BM (2022) Breast cancer segmentation recognition using explored DCT-DWT based compression. *Recent Patents Eng* 16:55–64
10. Sultana NN, Puhan NB (2018) Recent deep learning methods for melanoma detection: a review. In: Mathematics and computing: 4th international conference, ICMC 2018, Varanasi, India, Jan 9–11, 2018, Revised Selected Papers 4. Springer Singapore
11. Haenssle HA et al (2018) Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Ann Oncol* 29(8):1836–1842
12. Hekler A et al (2019) Superior skin cancer classification by the combination of human and (intelligence. *Eur J Cancer* 120:114–121
13. Fuadah YN et al (2020) Convolutional neural network (CNN) for automatic skin cancer classification system. *IOP Conf Ser: Mater Sci Eng* 982(1)
14. Höhn J et al (2021) Combining CNN-based histologic whole slide image analysis and patient data to improve skin cancer classification. *Eur J Cancer* 149:94–101
15. Patil R (2021) Machine learning approach for Malignant Melanoma classification. *Int J Sci, Technol, Eng Manag-A VTU Publ* 3(1):40–46
16. Sujaini H, Ramadhan EY, Novriando H (2021) Comparing the performance of linear regression versus deep learning on detecting melanoma skin cancer using apple core ML. *Bull Electr Eng Inf* 10(6):3110–3120
17. AlShourbaji I et al (2021) Early detection of skin cancer using deep learning approach. *Elementary Educ Online* 20(5):3880–3884
18. Hasan MR et al (2021) Comparative analysis of skin cancer (Benign vs. Malignant) detection using convolutional neural networks. *J Healthc Eng* 2021:5895156. <https://doi.org/10.1155/2021/5895156>
19. Bassel A et al (2022) Automatic malignant and benign skin cancer classification using a hybrid deep learning approach. *Diagnostics* 12(10):2472
20. Ghosh P et al (2022) SkinNet-16: a deep learning approach to identify benign and malignant skin lesions. *Front Oncol* 12:931141. <https://doi.org/10.3389/fonc.2022.931141>
21. Hossain M, Sadik K, Rahman MM, Ahmed F, Hossain Bhuiyan MN, Khan MM (2021) Convolutional neural network based skin cancer detection (Malignant vs Benign). In: 2021 IEEE 12th annual information technology, electronics and mobile communication conference (IEMCON), Vancouver, BC, Canada, 2021, pp 0141–0147. <https://doi.org/10.1109/IEMCON53756.2021.9623192>

# Prediction of Absenteeism at the Workplace: A Light Gradient Boosting Approach



Suresh Kumar Pemmada and Janmenjoy Nayak

**Abstract** The strength of a company or institution is linked to various elements, the most important of which is absenteeism. Employees on yearly leave or scheduled absences can be replaced by others; nevertheless, abrupt absences have an irrevocable negative impact. Employee absenteeism can harm an organization, resulting in less productivity and revenue. The objective of this research is to estimate absenteeism among prospective workers. In this study, the light gradient boosting machine (Light GBM) has been evaluated against bagging, XGBoost, k-nearest neighbor (KNN), multi-layer perceptron (MLP), and logistic regression (LR) using a well-known open-access dataset that is available in the UCI repository. In addition to the comparative analysis, we compared the efficacy of the proposed approach to earlier published research. The suggested Light GBM performs at 99.52% while bagging, XGBoost, MLP, KNN, and LR perform at 98.80%, 98.56%, 96.9%, 96.41%, and 93.8%, respectively. The proposed approach provides an efficient solution for organizations interested in knowing workers' behavior while recruiting and may lower the cost of paying inefficient or often absent personnel.

**Keywords** Absenteeism · Machine learning · Ensemble learning · Light gradient boosting

---

S. K. Pemmada

Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University), Visakhapatnam 530045, India

J. Nayak (✉)

Department of Computer Science, Maharaja Sriram Chandra Bhanja Deo University, Baripada, Odisha 757003, India

e-mail: [mailforjnayak@gmail.com](mailto:mailforjnayak@gmail.com)

## 1 Introduction

Companies strive to increase earnings while minimizing expenses to remain competitive in the market. Employees are critical components for firms from the ground up in pursuing goal attainment. Absenteeism is regarded as one of the most serious challenges for businesses, as it may increases costs and constitute an impediment to achieving corporate goals and objectives. According to Department of Labor research, over 2.8 million working days are missed annually due to employee absenteeism [1]. Regardless of what employers anticipate, regular absenteeism may diminish productivity and have a negative impact on the business and other elements. Absenteeism is a critical issue that must be addressed immediately by both the employer and the employee.

Their absence impacts the exploitation of resources and the employees' gross revenue. Employees may experience health issues and an increase in their physical and mental workload due to a fall in their gross income. Reduced employee absenteeism will enhance consumers' spending power and perhaps even the Gross domestic product (GDP) rate. Absenteeism is a serious issue, particularly for large organizations with many employees. Therefore, it is crucial to develop and use absence prediction systems in businesses that rely substantially on their people resources. Additionally, the absence rate in companies is a significant indication since it might ultimately have detrimental effects. It may act as a warning indication and emphasize the need for the business to take corrective action to address this issue [2]. Absenteeism at work is a significant indicator of a company's productivity and profitability. As a result, understanding employee absenteeism serves as the foundation for organizations of all sizes. Additionally, a company's trust may suffer when a service is interrupted, impacting how long it will take for the product to be delivered to clients.

Machine learning algorithms can examine current and historical data to find patterns and trends in employee absence. By doing so, employers and supervisors may acquire information regarding the chances of an individual being absent mostly in the future. This information may then be utilized to take greater concentration, such as giving extra training or providing incentives to minimize absenteeism. Furthermore, machine learning algorithms can detect employees who are at chance of being too absent, allowing companies to give extra resources and assistance to ensure that they stay productive and engaged. Despite its many benefits, machine learning has several drawbacks, including (1) Data collection (2) Time and resources (3) Result interpretation (4) High error-proneness, etc. Ensemble learning is a sort of machine learning in which multiple models are combined to provide more powerful and accurate predictions. Ensemble learning techniques boost accuracy by combining many models to reduce bias and improve the model's overall performance.

The key contributions of this work are as follows:

1. The Light GBM approach has been suggested in the study to identify workplace absenteeism.
2. We have utilized the "Absenteeism at work" data from the UCI machine learning library [3] to predict absenteeism in the workplace.

3. The suggested method performance has been validated using a variety of classification measures by comparing it with several machine learning models.

The remainder of the article is structured as follows: Sect. 2 provides a literature of workplace absenteeism. Section 3 investigates the experimental setup, separated into subsections such as dataset and preprocessing, environmental setup, performance metrics, proposed framework, and its architecture. Section 4 summarizes the findings and compares them to past studies. Section 5 summarizes the concluding observations and future scope.

## 2 Literature Study

This section discusses the use of machine learning methods to predict absenteeism. Dogruyol and Sekeroglu [2] employed three neural networks backpropagation (BP), long short-term memory (LSTM) neural networks, and radial basis function (RBF) to tackle the absenteeism at workplace, and then conducted comparison research between these techniques. Experimentation has been carried out using several training phases, such as 60 and 70%, and results were assessed using the mean squared error (MSE),  $R^2$  score, and explained variance (EV). The experimental results showed that the LSTM neural network had extremely high accuracy rates in predicting absenteeism at work. LSTM provided better outcomes than the other two neural network models. Nath et al. [4] utilized an approach for creating a web-based interactive tool for recruiting managers. “Absenteeism at work” data has been used in the simulation, and four machine learning models have been used to classify absenteeism classes, including support vector machines (SVM), multinomial logistic regression (MLR), random forests (RF), and artificial neural networks (ANN). They chose the optimal model based on validation score results such as precision, accuracy, F1-score, recall, and ROC-AUC. Then they incorporated it into their suggested web-based application for recruiting managers. MLR beat all other comparable approaches among all of these models.

Wahid et al. [5] utilized four major machine learning (ML) techniques, gradient boosted tree, decision tree, tree ensemble, and random forest, on absenteeism data from a courier firm in Brazil to identify employee absence. A variety of metrics have been used to evaluate the performance, including accuracy, sensitivity, true positives, false positives, true negatives, and false negatives. They observed that the gradient boosted (GB) Tree produced the best results, with an accuracy rate of 82%. Ali Shah et al. [6] presented neural network and deep learning algorithms for predicting employee punctuality at work. The suggested method’s effectiveness is evaluated against established machine learning approaches. The study revealed that deep neural network outperforms a single-layer neural network by 90.6% and SVM, decision tree, and random forest by 82%. Naganaidu et al. [7] investigated absenteeism data from a Brazilian courier firm and the causes of absence. The study’s primary goal is to predict three types of employee absenteeism using the multinomial

logistic regression model. Synthetic Minority Oversampling Technique (SMOTE) has been utilized to balance the classes, even though it impacted in accuracy class 3 accuracy has increased in terms of F1-score. Skorikov et al. [8] employed a variety of machine learning approaches, such as tree-based J48, zeroR, KNN, and naive Bayes, to analyze and predict absenteeism. The experimental findings reveal that the approaches can estimate absenteeism with more than 92% accuracy.

### 3 Experimental Setup

This section goes through the dataset specifics, the environmental setup, the performance measurements, and the proposed technique.

#### 3.1 Dataset and Preprocessing

Martiniano et al. [9] contributed the dataset for this work, which was downloaded from the UCI Repository [9]. It is generated using absence data from a courier company in Brazil between July 2007 and July 2010. The dataset has 740 samples and 21 unique attributes including the target variable performance assessment matrices. Data preprocessing includes combining some features and changing its type from one type to other based on the requirement. Table 1 presents data about the features reported in the dataset.

Some features in the dataset contain smaller values, such as 0–10, while others have large values, such as 100–1000, which may slow down the learning process. Data standardization is applied to improve machine learning algorithms' performance. During the dataset analysis, it is discovered that the moderate category contains more cases than the excessive category. Since most of the values are moderate, this might cause learning algorithms to be biased in favor of the moderate category. The SMOTE has been utilized to overcome the class imbalance problem.

#### 3.2 Environmental Setup

In order to conduct our research, we used Google Colab's environment with the scikit-learn, Light GBM frameworks. We utilized an 11th Gen Intel(R) Core(TM) i7 2.80 GHz processor, 16 GB RAM, Windows 11 (64-bit) 22H2 version, and an IRISx graphics card for this experiment. Other PYTHON packages, such as Imblearn, Pandas, and the Numpy framework, are used to examine the data further. In addition, an additional framework Matplotlib is used to visualize the data.

**Table 1** Dataset information

Attribute	Description/Remarks	Type	Distribution
ID	Employee identification number	Integer	–
Reason for absence	The cause of absence (from 1 to 28)	Integer	–
Month of absence	Employee's absence month	Integer	–
Day of the week	Sunday = 1, and so on up to 7	Integer	–
Seasons	Ranges from 1 to 4	Integer	–
Travel expense	Cost of transportation from home to work	Integer	Negatively skewed
Distance	Distance between home and work	Integer	Positively skewed
Service time	Service time in months	Integer	Negatively skewed
Age	Employee's age in years	Integer	Negatively skewed
Workload/day	Daily workload average	Integer	Positively skewed
Hit target	Employees' objectives	Integer	Negatively skewed
Disciplinary failure	It is 1 if the employee has a disciplinary failure record; otherwise, it is 0	Boolean	–
Education	The range is from 1 to 4	Integer	–
Children	Employee's number of children	Integer	Normally distributed
Social drinker	It is 1 if the employee is a social drinker; otherwise, it is 0	Boolean	–
Social smoker	It is 1 if the employee is a social smoker; otherwise, it is 0	Boolean	–
Pet	Employee's total number of pets	Integer	Positively skewed
Weight	Employee's weight (in kg)	Integer	Negatively skewed
Height	Employee's weight (in cm)	Integer	Positively skewed
BMI	Index of body mass	Integer	Positively skewed
Absenteeism time (target variable)	Absenteeism measured in hours	Integer	Positively skewed

**Table 2** Performance measures

Performance metric	Formula	Equation No's
Accuracy	$\frac{\text{True Negative} + \text{True Positives}}{\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative}}$	(1)
True positive rate (Recall)	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$	(2)
False positive rate	$\frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}}$	(3)
Precision	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$	(4)
True negative rate	$\frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$	(5)
F1-score	$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$	(6)
AUC	$\frac{\text{True Positive Rate} + \text{True Negative Rate}}{2}$	(7)

### 3.3 Performance Measures

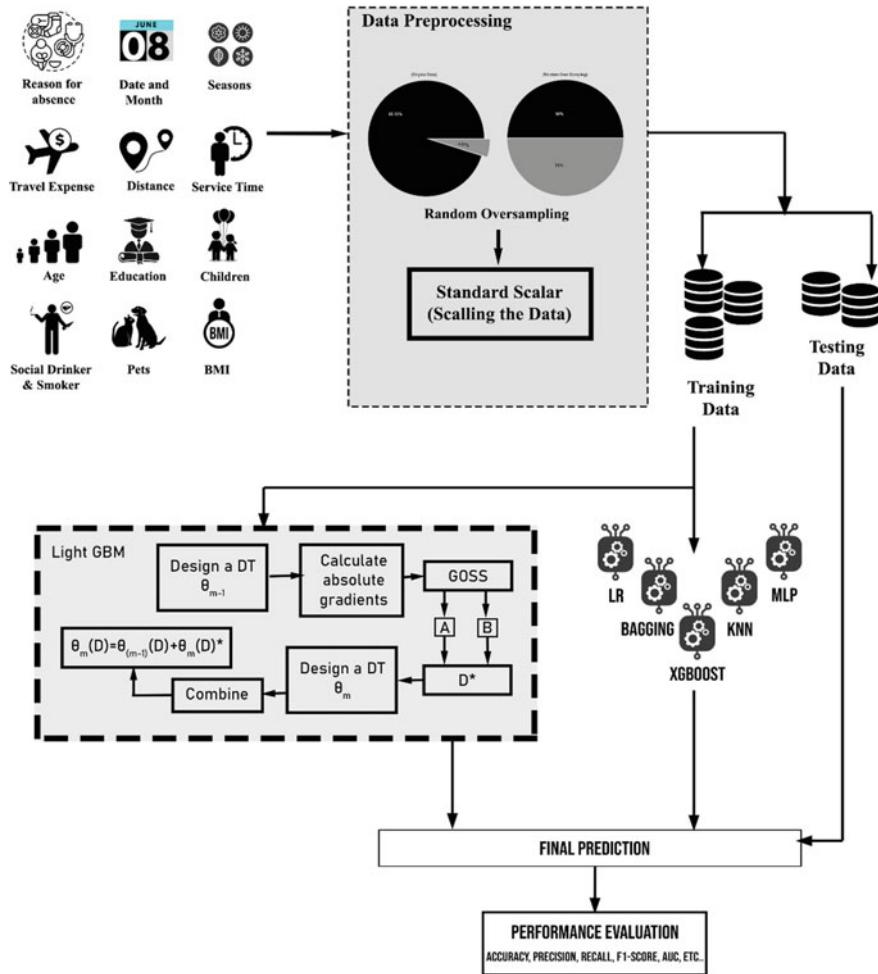
In this paper, the classification result for the proposed Light GBM is interpreted using the confusion matrix and obtained performance measures are listed in Eqs. 1–7 in Table 2 [10].

### 3.4 Proposed Method

This section describes the proposed intelligent method for detecting absenteeism based on tree learning algorithms and uses a light gradient boosting machine (Light GBM) framework. Light GBM starts by initializing the base learners before giving the LGBModel the input parameters for the number of iterations, loss function, and regularization frequency. The loss gradient is calculated using the provided loss function, and the predictions are set to zero. In each iteration, the base learners are fitted to the current loss gradient, and the predictions are updated by combining the predictions of the base learners. The loss gradient is also tweaked in the context of the new predictions. Finally, the model's final predictions are returned. The conceptual framework is presented in Fig. 1. As seen in the picture, we used random oversampling to balance the data. The data is split into training and testing, and then the proposed and alternative comparison techniques are tested. At the end, the key performance indicators are compared.

## 4 Result Analysis

Light gradient boosting, an ensemble learning technique, has been investigated to identify absenteeism in the workplace. In addition to experimentation, the research investigated several machine learning algorithms (LR, KNN, and MLP) and ensemble

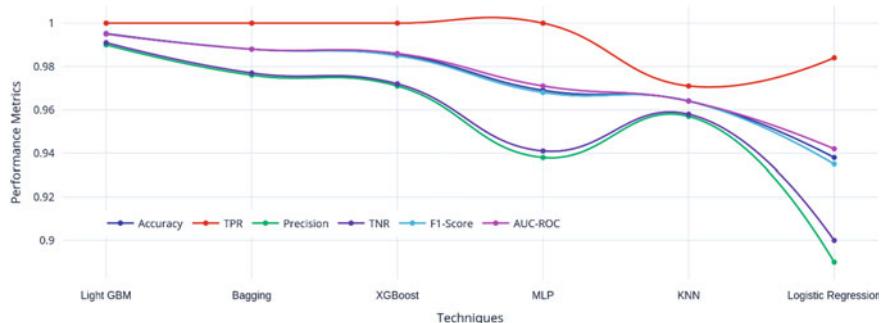


**Fig. 1** The suggested method's framework and comparison approaches

approaches (Bagging and Extreme Boosting) to illustrate the efficacy of the proposed Light GBM methodology. The suggested technique has been compared to several ML and ensemble learning (EL) methods using performance indicators such as true positive rate, precision, true negative rate, F1-score, false positive rate, and AUC-ROC. We divided the dataset into training and testing, with 80% of the dataset dedicated to training and the remaining 20% dedicated to testing. Table 3 compares the proposed method's efficiency to other ML approaches using different assessment measures.

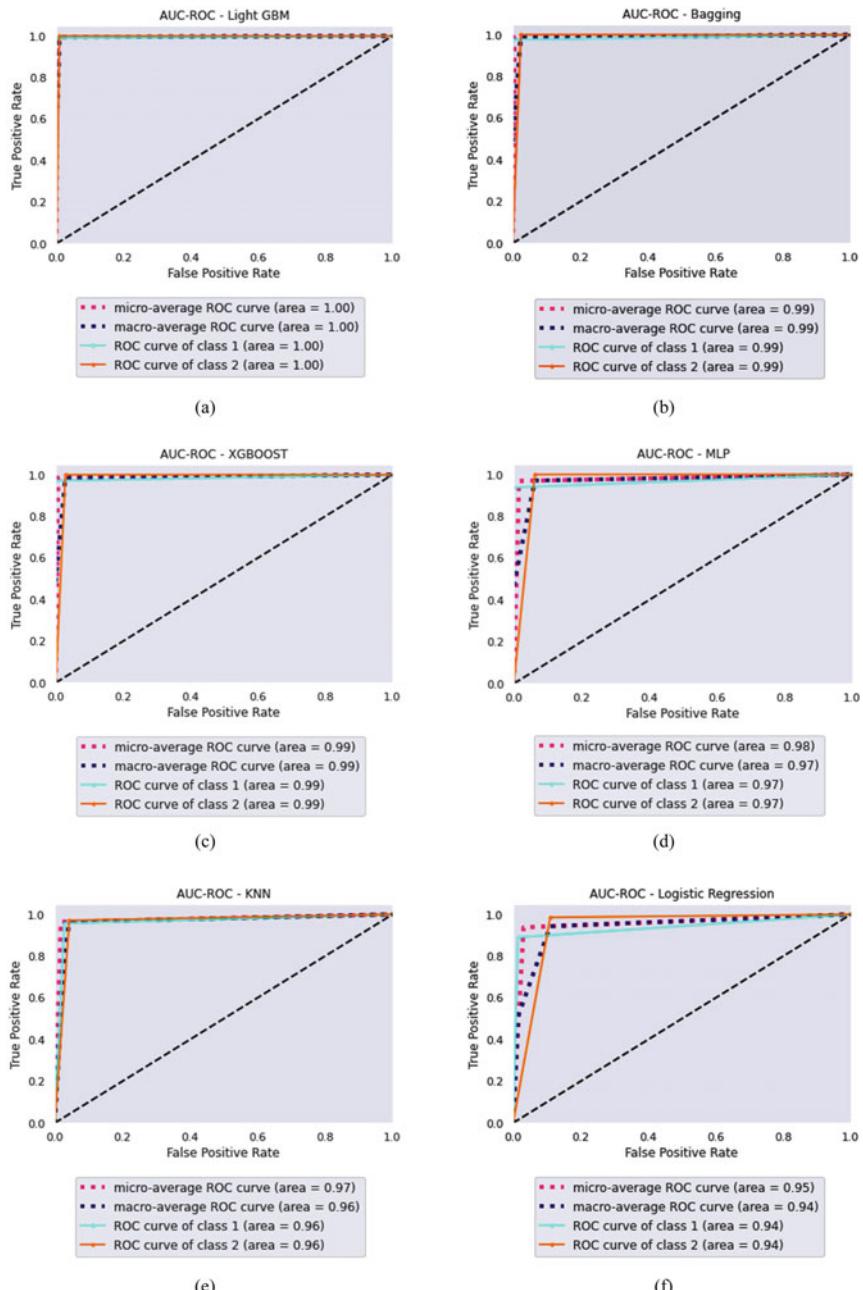
**Table 3** Performance of the proposed method and other compared ML and EL methods

Technique	Accuracy	TPR (Recall)	FPR	Precision	TNR (Specificity)	F1-score	AUC-ROC
Light GBM	0.9952	1.000	0.009	0.990	0.991	0.995	0.995
Bagging	0.9880	1.000	0.023	0.976	0.977	0.988	0.988
XGBoost	0.9856	1.000	0.028	0.971	0.972	0.985	0.986
MLP	0.969	1.000	0.059	0.938	0.941	0.968	0.971
KNN	0.9641	0.971	0.042	0.957	0.958	0.964	0.964
Logistic regression	0.938	0.984	0.100	0.890	0.900	0.935	0.942

**Fig. 2** The suggested and compared method's performances using several classification measures

The table clearly shows that the proposed Light GBM outperformed all ML, and EL approaches evaluated. The proposed Light GBM had the greatest accuracy of 0.9952, followed by bagging, XGBoost, MLP, KNN, and logistic regression, which have 0.9880, 0.9856, 0.969, 0.9641, and 0.938, respectively. In terms of true positive rate, Light GBM, bagging XGBoost, and MLP scored 1.00, with logistic regression scoring 0.984 and KNN scoring 0.938. Light GBM has the lowest FPR with 0.009, bagging and XGBoost have 0.023 and 0.028, respectively, while the other approaches have FPRs ranging from 0.042 to 0.1. Light GBM had the highest precision with 0.991, bagging, XGBoost, KNN, and MLP with 0.976, 0.971, and 0.957, respectively, followed by LR with 0.890. In many cases, it is seen that Light GBM scored quite well when compared to the other classification metrics, followed by bagging, XGBoost, KNN, MLP, and LR.

Figure 2 illustrates all assessment measures for the proposed method and different ML algorithms, including accuracy, TPR, precision, F1-score, TNR, and AUC-ROC. The suggested technique outperformed all the examined methods, especially Bagging and XGBoost, which performed well, followed by the proposed method. Figure 3a-f



**Fig. 3** ROC curve of **a** Light GBM, **b** Bagging, **c** XGBoost, **d** MLP, **e** KNN, **f** LR

show the ROC curves of the suggested technique and different ML and EL classifiers evaluated for various classes, as well as their scope in estimating the difference between them. In the case of the suggested technique, the micro average and macro averages of the curves and ROC curves of both classes is 1.0. In bagging and XGBoost, the micro and macro averages of the curves are both 0.99, and the ROC curve for both classes is similarly 0.99. The area of coverage, both in the micro average as well as macro average in the proposed approach, is higher in all methods, showing a greater ability to identify absenteeism in workload.

#### **4.1 Comparison of the Proposed Method with Previous Studies**

This section compares the proposed method's performance to past research performance. The proposed approach achieved the maximum accuracy of 99.52, whereas earlier studies ranged from 66 to 95. Table 4 shows a complete summary of the proposed approach and comparison methods.

### **5 Conclusion**

Absenteeism is a critical issue for businesses throughout their life cycle. Unexpected absence adds to the burden of other employees and reduces job efficiency. In order to adequately address employee absenteeism, which has a negative impact on a business's financial stability, the organization must foresee absence. Machine learning models can identify a potential class of workers based on job punctuality when recruiting. In order to address these concerns, we examined an open-access dataset from the UCI machine learning repository, experimented with Light GBM, and then compared several models (Bagging, XGboost, MLP, KNN, LR). The suggested Light GBM performs well compared to other proposals utilizing various performance measures. Additionally, we compare the performance of the suggested approach to the performance of previously proposed techniques (published research), and Light GBM surpasses all of them. In the future, latest ensemble learning algorithms will be implemented, and parameters will be modified using various optimization approaches.

**Table 4** Performance comparison of the proposed method and comparative method

Technique	Accuracy	AUC	F1-score	Precision	Recall	References
Decision tree	88.20%	0.801	0.829	0.864	0.882	[11]
KNN	91.31%	0.829	0.856	0.883	0.9131	
SVM	91.45%	0.851	0.869	0.892	0.9145	
Random forest	90.60%	0.843	0.871	0.886	0.906	
Neural network	95.60%	0.908	0.925	0.935	0.956	
MLR	0.932	0.885	0.915	0.937	0.932	[4]
SVM	0.887	0.87	0.898	0.916	0.887	
ANN	0.873	0.874	0.884	0.897	0.873	
RF	0.869	0.838	0.874	0.88	0.869	
Decision tree	0.797					[5]
Gradient boosted tree	0.824					
Random forest	0.804					
Tree ensemble	0.791					
DT	82.83	0.851	0.828		0.828	[6]
SVM	84.32	0.845	0.845		0.843	
RF	82.43	0.886	0.822		0.824	
A (zeroR)	85.5 ± 0.6%	0.5	0.79	0.74	0.86	[8]
A (Naive Bayes)	90.1 ± 2%	0.76	0.87	0.83	0.91	
A (J48)	89.7 ± 2.9%	0.74	0.87	0.83	0.91	
A (KNN-Euclidean)	88 ± 5.3%	0.74	0.87	0.83	0.91	
A (KNN-Manhattan)	88 ± 5.3%	0.74	0.87	0.83	0.91	
A (KNN-Chebyshev)	81.2 ± 5.9%	0.69	0.75	0.78	0.74	
B (zeroR)	85.5 ± 0.6%	0.5	0.79	0.74	0.86	
B (Naive Bayes)	66.5 ± 13.8%	0.8	0.82	0.85	0.8	
B (J48)	89.1 ± 4.3%	0.8	0.87	0.86	0.88	
B (KNN-Euclidean)	85.4 ± 1.3%	0.77	0.84	0.82	0.89	
B (KNN-Manhattan)	85.4 ± 1.3%	0.77	0.84	0.82	0.89	
B (KNN-Chebyshev)	86.9 ± 2.3%	0.67	0.87	0.83	0.91	
C (zeroR)	85.5 ± 0.6%	0.5	0.79	0.74	0.86	
C (Naive Bayes)	90.9 ± 1.6%	0.69	0.87	0.83	0.91	
C (J48)	90.9 ± 1.6%	0.69	0.87	0.83	0.91	
C (KNN-Euclidean)	90.9 ± 1.6%	0.69	0.87	0.83	0.91	
C (KNN-Manhattan)	90.9 ± 1.6%	0.69	0.87	0.83	0.91	
C (KNN-Chebyshev)	90.9 ± 1.6%	0.69	0.87	0.83	0.91	
Proposed method	<b>99.52</b>	0.995	0.995	0.99	1	

Bold is indicating that this is the highest accuracy among all

## References

1. Kocakülâh MC, Kelley AG, Mitchell KM, Ruggieri MP (2011) Absenteeism problems and costs: causes, effects and cures. *Int Bus Econ Res J* 8(5):81–88. <https://doi.org/10.19030/iber.v8i5.3138>
2. Dogruyol K, Sekeroglu B (2020) Absenteeism prediction: a comparative study using machine learning models. In: Advances in intelligent systems and computing vol 1095 AISC, pp 728–734
3. Andrea M, Ricardo Pinto F, Renato Jose S (2018) Absenteeism at work data set. <https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work>
4. Nath G, Harfouche A, Coursey A, Saha KK, Prabhu S, Sengupta S (2022) Integration of a machine learning model into a decision support tool to predict absenteeism at work of prospective employees. <https://doi.org/10.48550/arXiv.2202.03577>
5. Wahid Z, Satter AKMZ, Al Imran A, Bhuiyan T (2019) Predicting absenteeism at work using tree-based learners. In: Proceedings of the 3rd international conference on machine learning and soft computing, Jan 2019, pp 7–11. <https://doi.org/10.1145/3310986.3310994>
6. Ali Shah SA, Uddin I, Aziz F, Ahmad S, Al-Khasawneh MA, Sharaf M (2020) An enhanced deep neural network for predicting workplace absenteeism. Complexity, pp 1–12. <https://doi.org/10.1155/2020/5843932>
7. Naganaidu D, Mohd Khalid Z, Govindan S, Naganaidu D, Khalid ZM (2022) Prediction of absenteeism at work with multinomial logistic regression model. *Adv Appl Math Sci* 21(3):1479–1489
8. Skorikov M et al (2020) Prediction of absenteeism at work using data mining techniques. In: 2020 5th International conference on information technology research (ICITR), Dec 2020, no December, pp 1–6. <https://doi.org/10.1109/ICITR51448.2020.9310913>
9. Martiniano A, Ferreira RP, Sassi RJ, Affonso C (2012) Application of a neuro fuzzy network in prediction of absenteeism at work. *Iber Conf Inf Syst Technol Cist* 17–20
10. Rao BK, Kumar PS, Reddy DKK, Nayak J, Naik B (2021) QCM sensor-based alcohol classification by advance machine learning approach. Springer, Singapore, pp 305–320
11. Priyanka D, Nayak J (2020) Empirical analysis of absenteeism at work place using machine learning, pp 150–160

# Deep Neural Networks Scheme-Based Intelligent IDS System for Securing Internet of Vehicles



Ch. Ravi Kishore, D. Chandrasekhar Rao, and H. S. Behera

**Abstract** The Internet of Vehicles (IoV) refers to a network of vehicles that may exchange data and coordinate their movements via the use of sensors, wireless networking, and computer programs. With the advancement of technology, automation, and artificial intelligence, it is anticipated that the IoV will eventually replace conventional transportation networks. On the other side, this progress has expanded the potential of new intrusions. Intrusion detection is proposed in this research as a means of protecting the confidentiality of IoV network intra- and inter-vehicular communications. Deep learning (DL) techniques, namely Deep Neural Networks (DNNs) implemented to measure the Intrusion Detection System (IDS) efficacy in terms of recall, precision, F1-score, and accuracy. Further to achieve the optimal outcomes, our investigation shows that the suggested model's computational efficiency is better than that of the comparison ensemble ML models applied on bigger datasets. The suggested DNN approach achieved highest F1-score of 98.09%.

**Keywords** Machine learning · DL · IoV · DNN · IDS

## 1 Introduction

IoV is a novel idea formed by linked vehicles in Vehicular Ad-hoc Networks (VANETs) [1]. An Overview of Statista's 2021 report predicts that by 2030, there will be a total of 20.8 million autonomous cars and 146 million connected automobiles in the United States [2]. It will be more difficult to keep IoV networks safe

---

Ch. Ravi Kishore (✉) · D. Chandrasekhar Rao · H. S. Behera  
Department of Information Technology, Veer Surendra Sai University of Technology, Burla,  
Sambalpur, Odisha 768018, India  
e-mail: [cauchy9@gmail.com](mailto:cauchy9@gmail.com)

D. Chandrasekhar Rao  
e-mail: [dcrao\\_it@vssut.ac.in](mailto:dcrao_it@vssut.ac.in)

H. S. Behera  
e-mail: [hsbehera\\_india@yahoo.com](mailto:hsbehera_india@yahoo.com)

because of the increasing number of interconnected cars and the diverse nature of their network connections [3]. IDSs are used for the detection of malicious actions from inside a network or from the outside world that has bypassed other security measures. IDS may use either anomaly detection or the misuse detection technique. Misuse detection is used to identify known assaults based on their patterns, whereas intrusion detection is used to recognize any unusual activity in the network. It is hard to forecast an attacker's harmful requests to include them in a blacklist because of the attacker's ability to continually adapt and enhance their strategies [4].

An IDS that employs ML and DL to identify abnormalities in the IoV network is one of the most essential information security technologies. In several fields, numerous IDSs have been suggested. Object identification, Driver supervision, Autonomous Driving, Sensor Integration (i.e., radar and LiDAR), and attack and intrusion detection are some of the areas in which ML has been used in various parts of the IoV. Numerous anomaly based intrusion detection strategies based on ML have been proposed by researchers. In IoV networks, the advancement of new technologies and the increasing volume of online activity have resulted in the generation of greater volumes of multidimensional data and more sophisticated possible attacks, rendering shallow machine learning (ML) unsuitable for addressing the expanding security concerns.

DL is an improved version of ML that incorporates multi-layer neural networks to provide better outcomes. Neurons, which stand in for the mathematical calculation of the learning processes, link the many layers together [5]. DL has achieved success in several domains, including image and video identification, speech processing, processing of natural language automation technologies, and robotic systems, among others [6]. The usefulness of DL approaches in dimensionality reduction and classification problems has been shown. To identify threats more effectively, deep networks in the context of IDS learn from both conventional and unexpected traffic patterns created by an IoV network in the past. Deep networks are widely utilized to automatically minimize network complexity and explore data correlations [6].

The following are some of the most significant results from this study:

- We have suggested a DNN model which is an effective option for huge training datasets since it requires less computing power and can improve outcomes from massive datasets.
- There is a possibility that the training period may lengthen as a result of the complexity of the DNN model, however the result will be superior compared to traditional ML approaches.
- In contrast to state-of-the-art models, the suggested DNN model may quickly reach greater performance since it has already obtained the necessary information, including the important features along with weights associated with previously trained models.

The following categories describe the remaining parts. In Sect. 2, we compare and contrast the results of prior surveys that focused on DL-based IDS. Proposed DL architectures are outlined in Sect. 3. Afterward, Sect. 4 provides a summary of the dataset and simulation setup. Experimental research is described and compared

in Sect. 5, with a discussion of the comparative studies and the findings following. The research is concluded in Sect. 6.

## 2 Literature Survey

There are several research areas relevant to Intrusion Detection Systems in IoV networks that use DL and ML have been discussed below.

The UNSW-NB15 dataset has been used in the feature selection technique suggested by Kasongo and Sun [7] for intrusion detection using XGBoost. To test the efficacy of the IDS, the tests use a wide range of ML techniques (LR, SVM, KNN, DT, and ANN). XGBoost's feature selection approach is shown to help the DT achieve up to 90% accuracy in binary classification in UNSWNB15. As a feature selection approach for anomaly detection, recursive feature elimination (RFE) has been utilized in the research suggested by Ustebay et al. [8]. Ten key features have been extracted using RFE and fed into a Deep Multi-layer Perceptron (DMLP) classifier for usage on the CICIDS-2017 dataset. They have achieved an accuracy of 91%.

Rahul Vigneswaran et al. [9] have proposed DNN implemented on DARPA 1999 dataset. The hidden layer has ReLU as its non-linear activation function, the only two neurons that constitute the output layer are present there (Attack and Benign). They have achieved a 93% accuracy rate. Potluri and Diedrich [10] have suggested a DL-based DNN model with label-encoders and min–max normalization preprocessing technique implemented on NSL-KDD dataset, however, auto-encoder networks were implemented to train deep learning layers. They have obtained a maximum accuracy of 97.7% for the identification of DOS attack type, while probe detection accuracy reaches 89.8%. Using the KDD Cup 1999 dataset, Singh and Ahlawat [11] proposed an AI-based network IDS. In this system, preprocessing includes the use of PCA feature selection approach to minimize the dimensionality of original features and the Min–Max approach to normalize the input. In the context of ANN design, this research uses an ANN with forward propagation, backpropagation, and a MSE loss function. This model has achieved a 97.97% accuracy rate.

Liu et al. [12] have suggested a convolutional neural network (CNN) for network intrusion detection. The KDD Cup 1999 dataset have used to simulate this proposed model. This model has a 97.7% rate of detection accuracy. Using the Recurrent Neural Networks (RNNs) method and the NSL-KDD dataset, Yin et al. [13] developed a network detecting system. The results of this research may be split into two categories: 83.28% accuracy for attack and normal classification and 81.29% accuracy for multiclass classification. To identify malicious data, Aslahi-Shahri et al. [14] has designed a Genetic Algorithm (GA) using SVM. Selected features are fed into SVM and GA models. The researcher employed SVM for classification and regression issues. In this study, the authors used the KDD Cup 1999 dataset to achieve a detection accuracy of 97.3%.

With the use of the intelligent Internet of vehicles, Rani et al. [15] have designed an approach to predict and identify traffic congestion (IOVs). In this study, they suggest several different ML models, including Decision Trees (DTs), Random Forests (RF), Extra Trees (ET), and XGBoost. As compared to other models, XGBoost classification accuracy is 97.07%. To detect intrusions in IoV networks, Alladi et al. [16] have introduced DL-based misbehavior categorization techniques employing LSTM and CNN. An impressive 96.75% detection rate has been achieved by the suggested LSTM approach.

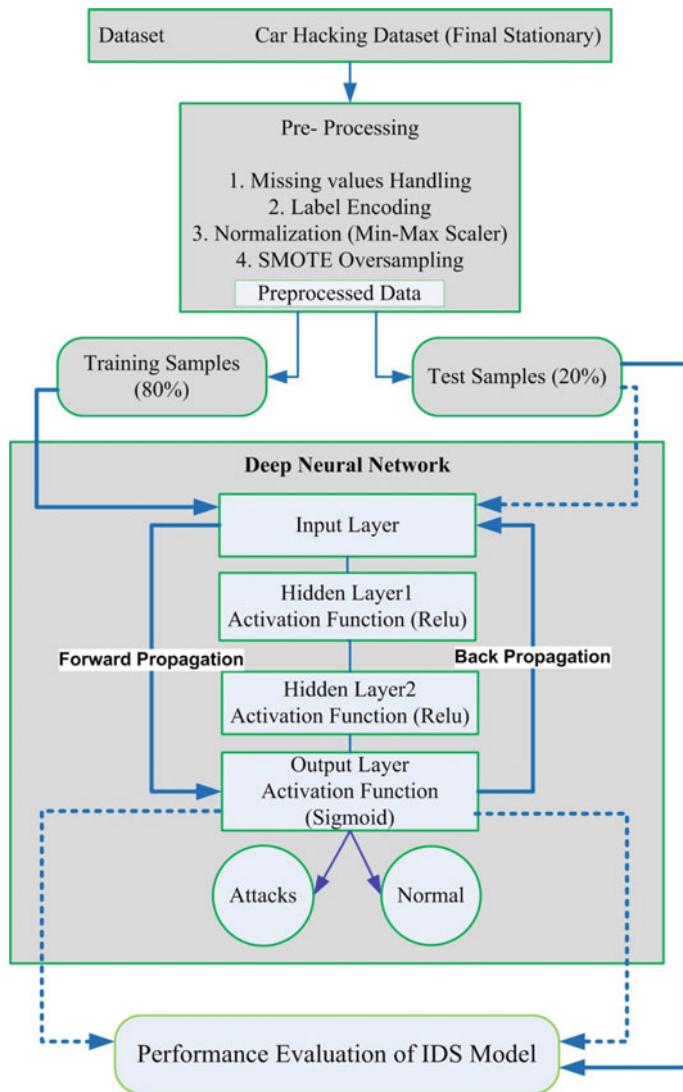
A novel approach Explainable Neural Network (xNN), has been developed by Aziz et al. [17] to classify attacks on an IoV network. The model performed well in tests measuring accuracy, recall, F1, and precision. The proposed model, xNN, clearly outperformed state-of-the-art models when a feature-scoring method has been adopted. The 98.01% accuracy achieved by xNN outperforms that of other popular models such as CNN (87%), LSTM (90%), and DNN (92.1%). Based on the VGG16 classifier DL model, Lin et al. [18] presented an intrusion detection model for IVNs to learn attack behavior features and classify threats. DoS, spoofing equipment, fuzzy attacks, and RPM in-vehicle communications are only some of the subjects addressed by the experimental dataset offered by the Hacking and Countermeasure Research Lab (HCRL). A false alarm rate of 97.82% has been obtained using the proposed approach.

The above literature study highlights that designing a standard IoV network security-based IDS framework has several traditional issues. (1) Vehicles lack computational and storage power to monitor and analyze activities. Hence, allocating processing and storage resources for the vehicle IDS is difficult. (2) A vehicle's topological communication environment is complicated, and weather and other circumstances affect driving. (3) Intrusion detection models generate many false alarms. In abnormal conditions, the conventional IDS model readily generates a high false alarm rate due to a vehicle's complex communication architecture. Consequently, when designing an IDS model for vehicles, we must consider the following characteristics. When deployed on platforms with restricted computational and storage capabilities, the developed IDS model can achieve reduced communication loads and utilize less storage space. It can adapt to the requirements of a highly dynamic topology and real-time processing power in automotive communications. Consequently, based on the study of the present security challenges of the vehicle network, an advanced DNN approach has been suggested for identifying abnormal behavior in the IoV network.

### 3 Proposed Work

One significant computational networks is a DNN, which is composed of numerous hidden layers, each having its neurons and mechanisms for connecting those neurons. The model presented in this research is the result of an algorithm using deep neural networks, and that process can be categorized into three distinct phases. First, the model's topology specifies the number of layers and the types of neurons that

comprise each layer as well as the interconnections between them. In the second, during forward propagation, artificial neurons incorporate with neural classifier and activation function. Finally, we have the loss function and optimizer for backpropagation. This paper describes a novel IDS that uses a DNN algorithm in conjunction with anomaly detection methods to identify assaults without violating users' privacy by inspecting the contents of their connections. The proposed DNN-based IDS in the IoV network is depicted in Fig. 1.



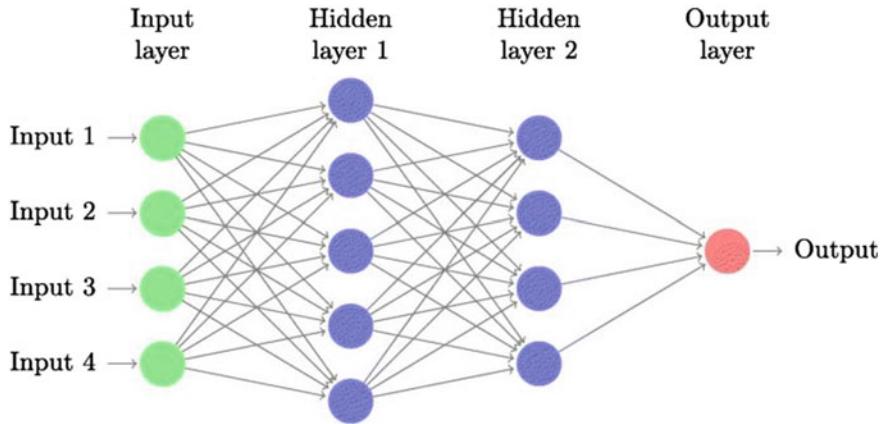
**Fig. 1** Architecture of proposed DNN-based IDS

### 3.1 Topology of DNN Model

**Input Layer:** Firstly, we have the input layer, which serves the neural network with some initial information. In our research preprocessed dataset's features are utilized to feed into a system with a 4-node input layer.

**Hidden Layer:** All calculations are executed in this layer, which lies between the input and output layers. The system has two hidden layers: a first layer with (64 neural nodes) and a second layer with (128 neural nodes).

**Output Layer:** It achieves the outcome (attack or non-attack). Every neuron in the input layer is interconnected to each neuron in the dense layer, and so on for all remaining layers. Figure 2 depicts the architecture of the DNN model. Table 1 is representing the model parameters setup for the training phase.



**Fig. 2** Architecture of DNN model

**Table 1** Parameter setup for proposed DNN

Parameters	Value
Number of input parameters	4
Number of hidden layers	2
Number of output parameters	1
Neurons in hidden layers	64, 128
Activation function (hidden layers)	Relu
Activation function (output layer)	Sigmoid
Dropout rate	0.03
Learning rate	0.01
Optimizer	Adam

### 3.2 Forward Propagation

Utilizing a perceptron classifier, the goal of forwarding propagation is to forecast outcomes (attack or normal). Multi-Layer Perceptron and Single-Layer Perceptron are the two forms of supervised learning that make up the perceptron. ANN used a multi-layer perceptron, and ANN serves as the foundation for DNNs. Equation (1) describes the fundamental equation of the perceptron.

$$Y = \sum_{i=1}^n X_i W_i + b \quad (1)$$

In the above equation, ‘ $n$ ’ represents the total number of neurons in a given layer, ‘ $x$ ’ represents the values stored in those neurons (values of the dataset), ‘ $W$ ’ represents the weights of the connections between them, and ‘ $b$ ’ represents the bias. All of the findings will be included in activation functions. Human brain research has given rise to the idea of an activation function, which states that a neuron becomes active whenever its activation probability reaches a certain value. This also limits the variety of possible outcomes. Sigmoid, softmax, tanh, and ReLU activation functions are often used. The ReLU activation functions were employed in this model’s hidden layers shown in Eq. (2), which are among the most prevalent. Sigmoid activation function represented in Eq. (3) has been implemented in the output layer. A Sigmoid activation function converts a prediction’s output to a discrete probability scale.

$$f(x) = \max(0, x) \quad (2)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

### 3.3 Backpropagation

To train a deep neural network, backpropagation is often used, since it allows for the modification of weights and biases. It has a loss function and optimizers in it. To find the optimal values for the model parameter, the loss function acts to reduce the value. Each model has its own unique set of parameters, and those values—known as weights and biases in the neural network—represent the model’s configuration. Cost function analysis allows for model assessment (loss function). Each parameter’s ideal value will be determined by minimizing the loss function. The optimal model parameter value can only be reached if the loss function is optimized (weight and bias). Optimizers are used to determine the optimal parameter value. Common examples of loss functions are Stochastic Gradient Descent, Batch Gradient Descent,

RMSprop, and the Adam algorithm. Several optimizers employed in this study were tested, and Adam was shown to be the most effective.

## 4 Dataset Analysis and Simulation Environment

This section describes the details discussion of the dataset and simulation setup.

### 4.1 *Details of Dataset*

Original Car Hacking Dataset (final stationary), which we used to evaluate the performance of our proposed DNN model. Hyundai Avante CN7 has been the competing vehicle. This final stationary dataset has a total of 1,270,310 samples, including 1,090,312 normal samples and 179,998 attack samples. Many examined models have a problem since there are more non-attack cases than attack examples, which causes the models to prefer the majority class. There is a class imbalance in the original dataset. Every ML model, including the proposed DNN model, uses the SMOTE to address the challenge of false distribution.

### 4.2 *Experimental Environment Setup*

For this research, Python notebook provided on Google Colab environment has been used along with GPU-powered servers, Keras, and Tensor Flow libraries. In our experiment, we used a system with an Intel Core i7 processor, 8 GB of RAM, and a Windows 10 operating system speed of 2.20 GHz (64-bit). Data analysis is performed using Pandas, Imblearn, and Numpy. Data visualization is performed using Matplotlib, and Mlxtend. While TensorFlow is a powerful open-source ML framework, Keras is a neural network library.

## 5 Result Analysis

The experiment is conducted on the suggested method in addition to a variety of standard ML classifiers, precision, recall, ROC-AUC, F1-score, and accuracy are used to validate the proposed strategy and other ML classifier techniques. The purpose of this research has been to examine the effectiveness of the selected models in various contexts, and the results of the ML metric analysis and the DNN model discussion are shown below. Since the F1-score helps achieve a suitable precision–recall ratio, we have assigned it the highest priority across all of our methods.

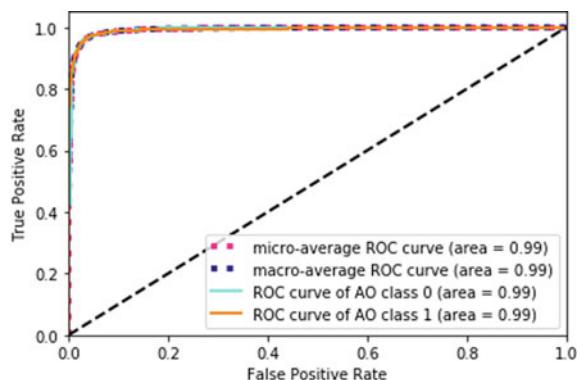
Several state-of-the-art ensemble ML methods, including Random Forest (RF), AdaBoost (Ada), Bagging (Bagging), Gradient Boosting (GBoost), and Extreme Gradient Boosting (XGBoost), as well as the suggested DNN based on deep learning neural network, are compared in Table 2. When compared to other state-of-the-art ensemble ML techniques, GBoost and Bagging provide the highest F1-score (90.59%) whereas AdaBoost has a poor F1-score of 79.51%. With an accuracy of 98.05%, a precision of 97.21%, a recall of 97.76%, an F1-score of 98.09%, and a ROC-AUC of 99.41, the suggested DNN model outperformed among all compared methodologies.

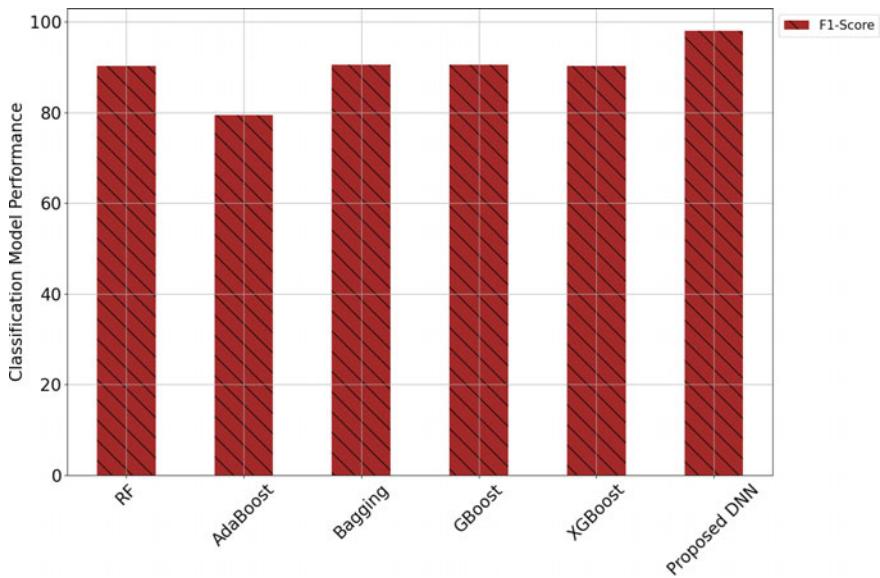
Figure 3 depicts the ROC-AUC analysis for the proposed deep learning-based DNN approach. In this particular research, DNN is significantly outperforming ensemble learning ML approaches in a significant way. In Fig. 4, the F1-score is used to contrast the investigated ensemble machine learning technique with the deep learning-based approach. When compared to traditional methods, the DNN-based strategy proposed here is superior, and it is validated as a reliable classifier. In Table 3, we compare the accuracy performance of the proposed model across several optimizers that were employed during its development. It demonstrates that, compared to other optimizers, the Adam optimizer provides the best performance.

**Table 2** Performance metrics comparison of ensemble ML models versus proposed DNN

Classification model	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC (%)	Accuracy (%)
RF	89.32	89.98	90.34	90.62	90.59
AdaBoost	78.33	78.76	79.51	79.54	79.53
Bagging	89.61	89.52	90.59	90.62	90.62
GBoost	89.61	89.52	90.59	90.62	90.62
XGBoost	89.13	89.32	90.23	90.48	90.49
Proposed DNN	97.21	97.76	98.09	99.41	98.05

**Fig. 3** ROC-AUC for proposed DNN





**Fig. 4** Comparision of F1-score for examined models

**Table 3** Accuracy analysis of different optimizers on the proposed DNN model

Model	Batch gradient descent (learning rate = 0.01)	Stochastic gradient descent (learning rate = 0.01)	RMSprop (learning rate = 0.01)	Adam (learning rate = 0.01)
DNN	96.75%	96.98%	97.21%	98.05%

**Table 4** Comparison of existing model with proposed model

Dataset	Suggested model	Accuracy (%)	Classification type	Refs.
Car Hacking Dataset-2020	XGBoost	97.07	Binary	[15]
	LSTM	96.75	Binary	[16]
	xNN	98.01	Binary	[17]
	VGG-16	97.82	Binary	[18]
	Proposed DNN	98.05	Binary	Our work

Table 4 demonstrates that the proposed model of this research study performs better than other approaches which have been implemented on the same dataset using a binary classification strategy.

## 6 Conclusion

In this study, we provide an architecture for monitoring IoV networks using a DNN-based deep learning approach. The model is trained on 80 percent of the data available in the Car Hacking Dataset 2020 (Final Stationary) using DNN techniques. ROC-AUC score, F1-score, and other performance measures are analyzed. Based on the evaluation measure, the suggested model can provide superior results when applied to bigger datasets. The effectiveness of the model may be studied in the future for various assaults by including various DL methods like the RNN and using a hyperparameter optimization methodology.

## References

1. Contreras-Castillo J, Zeadally S, Guerrero-Ibañez JA (2017) Internet of vehicles: architecture, protocols, and security. *IEEE Internet of Things J* 5(5):3701–3709
2. Placek M (2021) U.S.—Connected vehicles 2030. <https://www.statista.com/statistics/750113/us-connected-vehicles/>
3. Gupta DS et al (2022) Quantum-defended blockchain-assisted data authentication protocol for internet of vehicles. *IEEE Trans Veh Technol* 71(3):3255–3266
4. Hady AA, Ghubaish A, Salman T, Unal D, Jain R (2020) Intrusion detection system for healthcare systems using medical and network data: A comparison study. *IEEE Access* 8(106):576–584
5. Bengio Y, Goodfellow I, Courville A (2016) Deep learning. MIT Press. <http://www.deeplearningbook.org>
6. Hatcher WG, Yu WEI (2018) A Survey of deep learning: platforms, applications and emerging research trends. *IEEE Access* 6:24411–24432. <https://doi.org/10.1109/ACCESS.2018.283066>
7. Kasongo SM, Sun Y (2020) Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset. *J Big Data* 7(1):1–20
8. Ustebay S, Turgut Z, Aydin MA (2018) Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. In: 2018 International congress on big data, deep learning and fighting cyber terrorism (IBIGDELT). IEEE, pp 71–76
9. Vigneswaran RK, Vinayakumar R, Soman KP, Poornachandran P (2018) Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. In: 2018 9th International conference on computing, communication and networking technologies (ICCCNT), Bengaluru, India, pp 1–6. <https://doi.org/10.1109/ICCCNT.2018.8494096>
10. Potluri S, Diedrich C (2016) Accelerated deep neural networks for enhanced intrusion detection system. In: 2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA), Berlin, Germany, pp 1–8. <https://doi.org/10.1109/ETFA.2016.7733515>
11. Singh B, Ahlawat AK (2016) Innovative empirical approach for intrusion detection using ANN. *Int J Innov Res Comput Sci Technol (IJIRCST)* 4
12. Liu Y, Liu S, Zhao X (2017) Intrusion detection algorithm based on convolutionalneural network. ICETA. <https://doi.org/10.12783/dtetr/iceta2017/19916>
13. Yin C et al (2017) A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* 5:21954–21961
14. Aslahi-Shahri BM et al (2016) A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Comput Appl* 27:1669–1676
15. Rani P, Sharma R (2023) Intelligent transportation system for internet of vehicles based vehicular networks for smart cities. *Comput Electr Eng* 105:108543

16. Alladi T et al (2022) A deep learning based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems. *Digit Commun Netw.* <https://doi.org/10.1016/j.dcan.2022.07.006>
17. Aziz S et al (2022) Anomaly detection in the internet of vehicular networks using explainable neural networks (xNN). *Mathematics* 10(8):1267
18. Lin H-C et al (2022) Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks. *Electronics* 11(14):2180

# Low-Memory Pedestrian Detection Using Binarized Neural Networks



Mainak Bandyopadhyay and Rakesh Baral

**Abstract** Pedestrian detection is a very active topic in study because of its wide applications in fields such as automotive, security, and surveillance, as well as diverse use cases like traffic control and intruder detection. Although there has been great progress in this area of research, there are still many obstacles to overcome in terms of real-time utilization. In recent years, deep learning and convolutional neural networks have provided significant improvements in terms of accuracy, beating prior standard image processing algorithms. However, they face challenges and difficulties while installing for the very reason of large size of the models. Addressing this problem statement, binarization of the parameters and lessening intermediate operations could yield encouraging outcomes. Previous works with regards to quantization of weight values and activations have been proposed for the development of applications for the less power hungry applications. In this paper, an architecture for the pedestrian recognition and localization has been developed using quantization processes, with the goal of making them simple to implement in embedded devices or portable assisted devices. This model will suffer from accuracy degradation in comparison with the existing deep networks, but it will be more flexible to utilize in multiple modes. This paradigm will eliminate the need for external servers for computation, making the systems self-contained and secure. In addition to it, it will consume less power which would be beneficial for the healthy battery life. Finally, diverse datasets have been evaluated, and statistical analyses suggest its feasibility for utilizing it.

**Keywords** Pedestrian detection · Binarized neural network · Binarized DenseNet · Pedestrian localization

---

M. Bandyopadhyay (✉) · R. Baral

School of Computer Engineering, KIIT Deemed to Be University, Bhubneswar 751024, India  
e-mail: [mainak.bandyopadhyayfcs@kiit.ac.in](mailto:mainak.bandyopadhyayfcs@kiit.ac.in)

R. Baral

e-mail: [2064004@kiit.ac.in](mailto:2064004@kiit.ac.in)

## 1 Introduction

Artificial Intelligence and Computer Vision, in particular, assist in analyzing any type of visual content in the form of a frame or sequence of frames in order to perform the necessary steps ahead of time. Pedestrian detection is one of those significant research areas that has been raging for several decades for its various applications. In the last couple of years, there has been significant improvements in terms of both efficiency and accuracy due to the surge of interest in the deep convolution networks. Both object detection and pedestrian detection share quite similar architectural design and analysis. The biggest advantage of using deep network solutions is there is no need to use any specific handcrafted modules to extract features from an image in the classification problem. For the localization task, each region proposal is analyzed to check whether it contains any human alike shape or not. For both of the tasks above, the classifier should be properly trained with positive and negative image datasets in different scenarios. But all of the available solutions are deep learning-based solutions which are only be useful for high-end devices and require too much power to process. The heavy computation due to full-precise operations give a very decent accuracy but fail to provide on-device solutions, e.g., AlexNet [1] has about 61 M floating point parameters which at a minimum takes 249 MB of memory to classify a single image. Even DenseNet121 which is the smallest of the DenseNet category having around 7 M parameters take around 27 MB of the memory with 925 MB of additional processing unit for the required matrix operations. Integrating all the state-of-the-art recognition and detection systems to the smart devices provides significant results in solving the real-world applications. Focusing on low latency use cases which ultimately consume low power and increase the battery longevity is suitable for real-time assisted devices. This can be achieved by reducing the run time memory and accelerating the inference time taken. In addition to that, the precision of the parameters can be reduced or quantized to fit them in the low computational devices like mobile and embedded devices. Compressing the parameters to binary values can achieve up to 32 times low memory usage and can speed up to 58 times faster with implementation of compatible hardware [2], but somehow impacts the accuracy. There should be proper trade-off between all the impacting factors and maintaining the rich information flow throughout the network.

## 2 Literature Review

This section describes various existing strategies in the domain of pedestrian detection and the progress in the area of binarized neural networks. The area of pedestrian detection differs from the human face recognition, rather it treats human alike shape as objects and requires object detection like solutions. Tome et al. [3] described the standard technique to design a pipeline for training and inference purposes, which is quite similar to the object detection architecture. The traditional image processing

methodologies require extracted feature maps prior to the machine learning classifier model applied to it. Among all the traditional algorithms, the most initial remarkable one is the HOG-based detector. The HOG, SURF, and SIFT descriptors and its hybrid variants are earlier used to train the model detection, and feature approximation approaches can also accomplish scale invariance and nonlinearity. The steady advancement of CNN-based detectors does not require any specific layers to extract features from frames, rather convolution layers do so internally. Deep networks such as VGGNet, DenseNet, and GoogleNet provide cutting-edge accuracy and can be very much effective against a wide range of camera angles, occlusions, textures, and colors [4]. The introduction of R-CNN and all its variants [1] gives out a set of focused areas of the frame with the application of region proposals where the likelihood of finding the object is more and achieved top performance on existing pedestrian datasets. Moreover, the approaches like non-maximal suppression (NMS) and adaptive non-maximal suppression [5] add more layers in the post-classification and regression methods to meet the cutting-edge performance on occlusions like constraints. Advanced pedestrian detectors [6] achieve improved performance by stacking together multiple CNN-based pedestrian detectors. As a result, CNN-based generic architectures become expensive and challenging to implement in the on-device or in-situ processing requirements. Fernando Cladera Ojeda et al. [7] offer a unique architecture in which the deployment of a binary neural network can significantly reduce complexity and can be a feasible solution. Given the constraints of CNN-based architectures, utilizing a binarized neural network (BNN) may be a viable solution. Although this algorithmic technique saves computing time, memory space, and power consumption, it confronts significant challenges in terms of training and optimizing the weight parameters. Recently, improved algorithms have been presented in recent years to ensure the smooth operation of the BNN. In [8, 9], significant results in terms of directions for binarizing a network and to explore the other possibilities have been proposed. The work concentrates on the quantization of weights and the activation function outputs in the range of 0–1, as a result of which it may be readily binarized. Intuitively, the approximation of complete precision neural networks aids in the reduction of all restrictions such as power and memory. Since binary weights cannot be differentiated, concepts like straight through estimators (STE) come into play which acts like dropout layers in this class of neural networks. In [10], all the existing optimizers are classified into four categories and also implemented in the quantized neural networks. With the efficient use of all of these methodologies, a BNN may be applied to construct intelligent smart systems and could be beneficial for the use in embedded systems.

### 3 Motivation and Contribution

The most efficient models for pedestrian identification make use of extremely dense deep neural networks. But all the existing deep learning-based solutions go through below technical limitations.

- The overall number of operations required inside a deep neural network is determined by the total number of full exact parameters employed and the depth of the entire network. It necessitates a larger cache size and a greater number of memory operations. As a result, most deep network models consume more power and take up more space.
- Secondly, the number of parameters are very large, and each fully precised parameter consumes 32 bit of memory space. In current hardware with adequate accuracy, deep networks can be optimized because the majority of them do not contribute to the final outcome. Essentially, these networks are subject to over-parameterization and leads to very large layer operations.

In this paper, a simple method for approximating existing popular CNNs for usage in the domain of pedestrian detection is demonstrated. The DenseNet121 architecture has been implemented for the advantages it has for the object detection, and concepts from the binary neural network have been used for the approximation and quantization operations. The deep network employs quantized weights rather than actual weights, and the layer actions are binarized. This will undoubtedly reduce accuracy figures, but at the expense of fewer memory operations. This architecture can be used when developing low-weight applications for many types of on-device calculations in the real time.

### 4 Related Prerequisites

This section provides brief explanation about primary implementation principles of the binary layers and binarized version of the deep network.

#### 4.1 *Binary Quantization*

Deep learning quantization is the approximation of a neural network using smaller bit numbers rather than full-precision floating point integers [11, 12]. Deep learning quantization is the procedure for limiting the existing values' range to the specified range. It improves the efficiency of networks and is mostly utilized for on-device calculations. Binary quantization is a subset of it in which the intermediate weight values are transformed to binary bits to reduce the amount of running memory required and the number of operations required. The most frequent approach is to

use the sign function for the binary activation function and binary weight conversion provided in Eq. (1).

$$\text{sign}(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases} \quad (1)$$

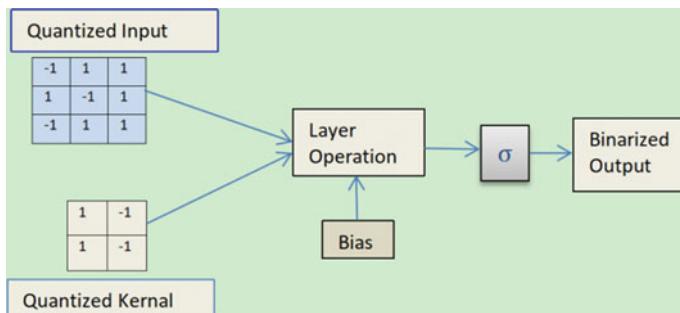
The output of this function is a binary number, with  $x$  being the real-valued variable.

## 4.2 Quantized Layer Operation

The layers in the normal deep networks can be quantized both in terms of weights and kernels which are known as weight quantizer and kernel quantizer, respectively, as shown in Fig. 1 [2]. The quantized layer operations exclusively perform binary operations, and even the matrix multiplication operation is performed using bit-wise operations. It would benefit in terms of computational complexity and storage requirement as for storing it will not require 32 bits for each element. Each layer operations are modified as provided in Eq. 2 to accommodate the quantized binary values.

$$y = \sigma(f(q_{\text{kernel}}(w), q_{\text{input}}(x)) + b) \quad (2)$$

Quantization of operations decreases the information flow between layers. One of the reasons for this is due to loss of information from switching from full precision to binary. As suggested in [11], normalization minimizes the difference between the full-precisioned network and the binarized neural networks. Each layer operation or convolution operation should be followed by a batch normalization layer.



**Fig. 1** Each node operation is shown where every full-precisioned value is quantized to binary values before the operation

### 4.3 Straight Through Estimator

The network adds some noise to the weights and activations by thresholding the weight value, which works as a regularizer for the network. With  $\text{sign}(x)$  function as the activation function the gradient descent will not function and the gradient will not pass in backward phase. The concept of a straight through estimator (STE) is introduced for the back-propagation rescue. STE is an empirical method of maintaining the gradient approximation [13]. This method is prevalent in all quantized neural networks and works according to Eqs. 3 and 4.

$$\text{Forward pass} : r_0 = \text{sign}(r_i) \quad (3)$$

$$\text{Backward pass} : \frac{\partial c}{\partial r_i} = \frac{\partial c}{\partial r_o} 1_{r_i \leq 1} \quad (4)$$

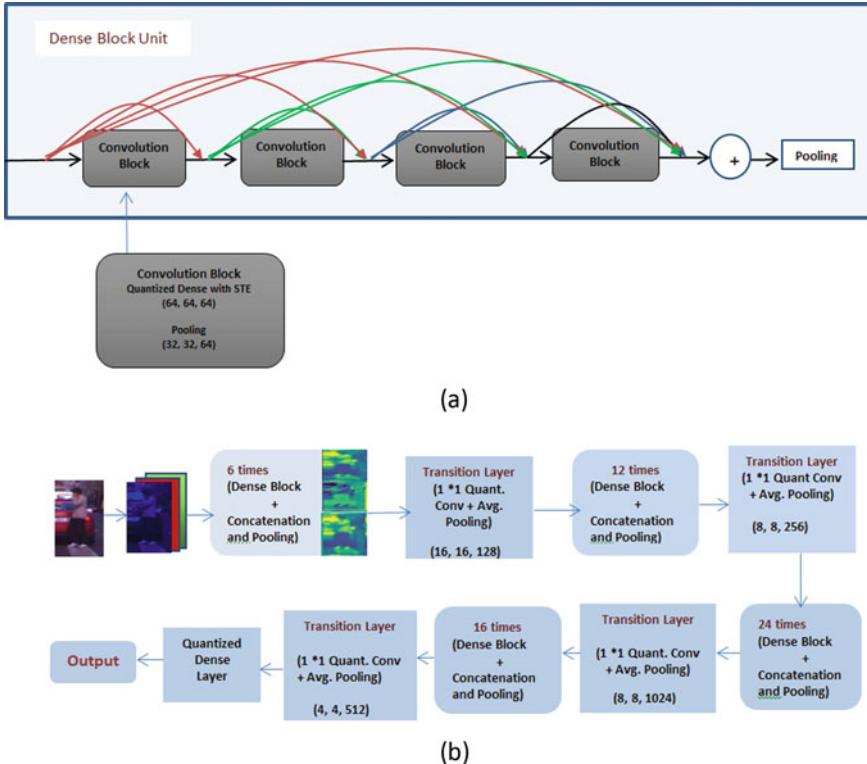
Here ‘c’ is the cost function and ‘r’ signifies the intermediate parameters. By applying this function to the gradients, it behaves similar to the  $\tanh$  replacing the sign function in the backward pass. This function allows the modification if it ranges between  $-1$  to  $+1$  and cancels the gradients if it gets too large.

### 4.4 Binarized Version of Deep Network

Empirical works carried out suggests that binarized versions of deep learning architectures can be trained using three approaches (i) fully from scratch, (ii) fine tuning a full-precision architecture, and (iii) clip. Out of these three, training from scratch obtains better results than other. Some of the modifications suggested for binarization of full-precision networks to maintain information flow in the network includes:

- (a) Use of shortcut connections to preserve the information loss in binarization by increasing the information flow in the network.
- (b) Eliminating the bottlenecks or increasing the number of kernels in bottleneck parts. In case shortcuts are not introduced, important layers should be kept full precision, e.g., first and the last convolution layer.

DenseNet architecture is a popular architecture and can be binarized [14]. DenseNet and its derivatives use shortcut connections to approximate the full-precision counterparts by concatenating the block’s input to the output. The current work introduces a binarized variant of existing DenseNet121 shown in Fig. 2 for solving the pedestrian detection problem statement. Due to binary layers in the dense blocks, it is somehow challenging to achieve satisfactory performance. Manipulating the growth rate factor ( $k$ ), i.e., channels in dense blocks and numbers of dense blocks, achieves better performance. One such setup is to double the number of dense blocks and halve the growth rate factor. Binary downsampling with low reduction rate or no



**Fig. 2** **a** Binarized dense block. **b** The binarized version of DenseNet architecture used in the proposed approach. The quantized convolution with STE is implemented instead of using the regular convolution operation

reduction is also an arrangement to improve accuracy. However, this adds additional computational costs during the training as the network width gets increased in every concatenation operation.

## 5 Proposed Architecture

In general, the proposed system is designed with the requirements of light weight and real-time processing in mind. The system is developed to meet the needs of portable and power-efficient embedded mobile systems in terms of efficiency and precision. The previously mentioned common framework has been divided into two independent modules: detection and localization. Each module has the ability to be individually incorporated into IoT devices, depending on the requirements. By avoiding running the localization module needlessly, the system attempts to save power consumption. Following are the functional description of the modules.

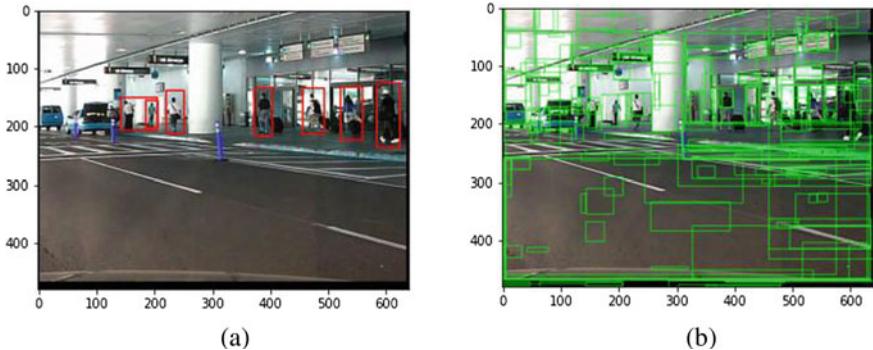
## 5.1 Detection Module

The frames collected from a streaming video at a predetermined regular interval are used to identify the frames as pedestrian or non-pedestrian. Because the image is immediately supplied into this categorization model, this module is developed utilizing binarized deep learning modules, which results in very low latency. This module will be trained with positive classifiers such as normal straight human-like shapes and negative classifiers such as diverse backgrounds and terrains. Only those frames that contain one or multiple pedestrian are selected as output in this module.

As a result, the processing cost for the next stage is decreased by deleting frames with no human-like features. However, the output will be imprecise, and the pixels that contribute to human detection will be scattered rather than pin pointed as in a traditional DenseNet.

## 5.2 Localization Module

This module uses a binarized deep learning model to output bounding boxes around pedestrians. Region proposals are extracted in the frames as part of the preprocessing. The region proposals are those locations where there is a very possibility of discovering the object. Partially human shapes, occluded images, and images with diverse backgrounds, such as packed streets, clear roads, and sky views, are all possibilities for area ideas. The region proposals are determined using a hard-coded selective-search algorithm that takes into account texture, color, pattern, and other factors shown in Fig. 3. When the IOU score is compared to the ground truth boxes in the input data, the bounding boxes are labeled positive or negative. These rectangular-shaped boxes are fed into a deep learning network, which determines whether or not the object inside is a pedestrian. In the end, the non-maximal suppression (NMS) class of techniques is used to remove many bounding boxes around a single item. For finding the best bounding box among all the region proposals, an approach inspired from the NMS class of algorithms has been used as shown in Fig. 5. This works well in case of single pedestrian detection but fails to identify the correct one in case of crowded areas where multiple bounding boxes are overlapping having multiple persons residing nearby. There is a room of improvement of the observed miss rate by using the advanced version of NMS like adaptive NMS, etc. Here, among the overlapping bounding boxes, i.e., those having the IOU scores more than 0.2, the one which is having the maximum probability score in the output is taken as the final bounding box and rest are suppressed. Because there is a lot of preprocessing and post-processing to be done, this module will be heavier than the detection module. When this module is run in serial, it will approximate the pedestrian's path, which can be useful in a variety of situations.



**Fig. 3** **a** Ground truth bounding boxes given as the input and, **b** bounding boxes generated as region proposals by the selective-search algorithm

## 6 Experimental Setup and Results

The below experiment is performed using the benchmark datasets which are as follows.

- The classification experiment was carried out with the INRIA Person Dataset, which is one of the most widely used datasets for pedestrian detection. In this experiment, a total of 2000 positive and 3000 negative images with varied zoom angles and resolutions are trained and tested in the 0.7/0.3 ratio.
- CVC01 is a pedestrian classification dataset where randomly captured frames in two categories such as positive and negative are trained and tested. Here, total around 1400 images of person and 600 randomly taken negative images having street views used in the experiment.
- CVC02 consists of total 1016 images of pedestrians along with their corresponding mirror images that are fed as the positive images and 7650 negative images in different terrains which are given as negative images.
- The popular localization dataset Caltech is also tested where the images with bounding boxes are trained after the preprocessing. Only 10 percent of data of Caltech Set01 and Caltech Set02 datasets with around 9000 images are trained with positive and negative bounding boxes maintaining the ratio of positive to negative boxes as 1:6. Ultimately, the bounding boxes are predicted using the quantized deep network.

The experiments are performed using CPU with 16 GB RAM and the popular Python packages. The functions to build the neural networks are imported from Keras–Tensorflow, and for binarizing them, Larq APIs are used. For each dataset, the ADAM optimizer is used for the advantage of adaptive learning it has and the effectiveness in the area of binarized neural networks as found before [11]. The training is done with 0.7 percent of total dataset and rest are used for the testing purpose. The total number of epochs is set to 100. The number of simulations considered here is 20.

## 6.1 Performance Metrics

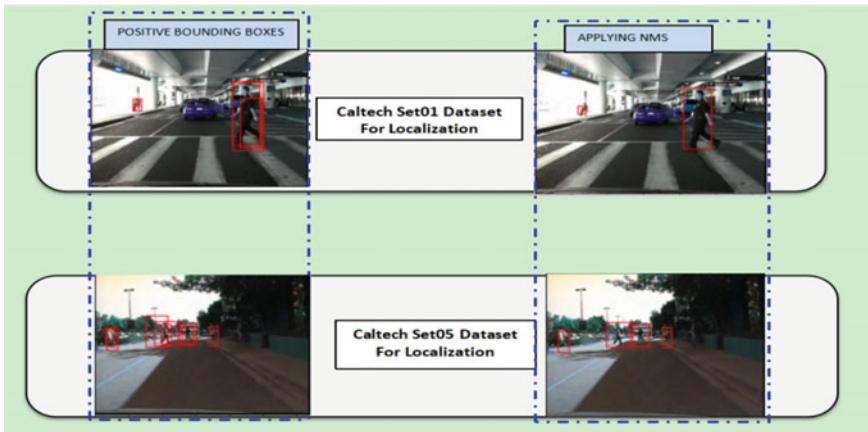
To evaluate the robustness of the binarized model, the commonly used evaluation metrics for real-time pedestrian detection such as average precision (AP), miss rates, and memory footprint were calculated.

- **Mean Average Precision (mAP):** Average precision is considered as the measurement of the accuracy here, and it is the area under the precision–recall curve. It basically shows the number of correctly classified classes in an evenly distributed dataset like INRIA and CVC01.
- **Miss Rate:** In an unbalance dataset like Caltech, the number of negative classes is more as different type of textures, backgrounds, and scenarios are taken in the dataset. Hence, precision cannot be a good metric to measure the robustness of the model. In such datasets, the focus is more toward how many human alike shapes are detected and how many are missed. Hence, miss rate can be an appropriate measurement. It is the ratio of the false negative to the total number of positive frames, i.e., total pedestrians in the input images.

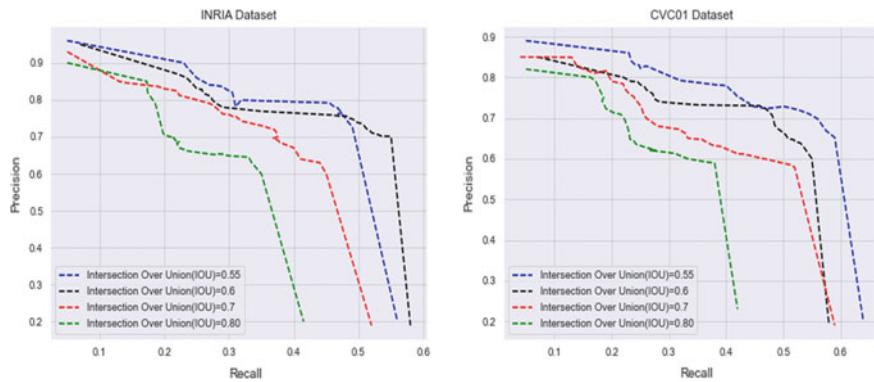
## 6.2 Observations

The section provides some observations obtained from the simulation results. Accuracy metric, total number of parameters, and memory space are considered as a parameter for comparative evaluation of full precision network and binarized network. To achieve a better visual clarification, the intermediate picture outputs were added as shown in Fig. 4. The model size indicates that it is compatible for an embedded device. Thus, possibly can be directly implementable in mobile device applications. The convolution operation is performed using the multiple accumulate (MAC) function, which is the extremely expensive operation. As provided in Table 1, 90% of the matrix multiplication operations are binarized in the binarized model.

By binarizing the weights and the kernel values provided in Table 2. Thus, the binarized model performance is better in terms of computation speed than full precision network. There are many formats available for small devices like mobile phones, embedded devices, and other IoT devices such as TFLite. Here, binarizing the weight values reduces the model size to 2.17 MB which is three times smaller and thus lighter than the full precision network model. It enables on-device requirements to be met without the use of external servers and uses little power suitable for low-weighted applications. The total operations include all the multiplication and addition operations performed throughout a single epoch of the deep network propagation. Hence, the total number of parameters will remain unchanged in a completely trainable deep network as the parameters are only quantized during the run time. This will not have any effect with the overall performance in comparison with the corresponding full-precised version of the neural network. Already mentioned as the task is related to binary classification, the average precision and miss rate for each dataset were used



**Fig. 4** Localization module output which gives out the positive bounding boxes and applying NMS to it gives the final output



**Fig. 5** Precision–recall curve with the various setting of IOU

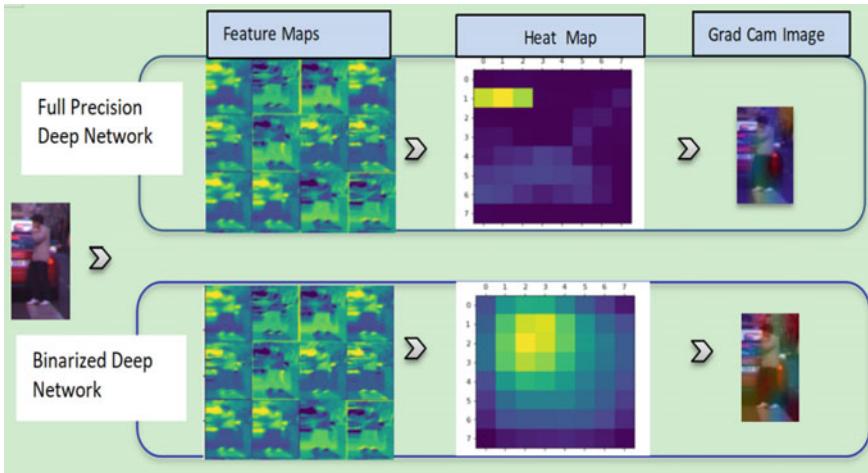
**Table 1** Comparison of various parameters in regular DenseNet, MobileNetV2 with the binarized model

	DenseNet	MobileNetV2	Binarized deep network
Float 8 weight size	6.64 MB	4.04 MB	928.46 KB
TFLite model size	6.64 MB	4.04 MB	2.17 MB
Total trainable parameters	6.88 M	4.23 M	6.88 M
Ratio of MACs that are binarized	0	0	0.9

**Table 2** Performance in terms of average precision and miss rate for the balanced and imbalanced datasets, respectively

Dataset	Model	Mean average precision (mAP)	Miss rate (MR)
INRIA	DenseNet121	91.7%	–
	MobileNet + SSD	84%	–
	Binarized DenseNet121	80%	–
CVC01	DenseNet121	92%	–
	MobileNet + SSD	81%	–
	Binarized DenseNet121	76%	–
Caltech Set01	DenseNet121	–	0.12
	MobileNet + SSD	–	0.22
	Binarized DenseNet121	–	0.24
Caltech Set05	DenseNet121	–	0.1
	MobileNet + SSD	–	0.19
	Binarized DenseNet121	–	0.21

to assess the binarized model’s efficacy. The average precision infers the number of successfully recognized frames to the total number of frames being tested in a balanced dataset. The miss rate infers the ratio of pedestrians who were missing to be detected to the total number of pedestrians in the localization module. The average accuracy and miss rate have been reported in the table after simulating the model 20 times on each dataset. In the INRIA dataset, the accuracy value is around 80%, whereas in the CVC01 dataset, it is 76% as noted in Table 1. Provided in [15], the state-of-the art spatio-temporal DenseNet architecture achieves around 84% of average precision in real urban traffic environments. The state-of-the-art MobileNet + SSD network which is widely used and designed specifically for the mobile devices achieve 80.04% of average precision in the real-time scenarios [16]. Because of the low accuracy weight values, the model is unable to detect objects in the presence of occlusion, blur, or poor brightness, among other things. In addition to that from the Heatmap and the GradCam image provided in Fig. 6, it can be noticed that the binarized neural network output is less precised compared to the full-precised network. As the dataset is uneven, the performance of the localization module is judged by the miss rate. In Caltech Set01 and Caltech Set05 datasets, the miss rate is roughly 0.24 and 0.21, respectively, which means that 24% in Caltech Set01 and 21% in Caltech Set05 of all pedestrians go unnoticed. This occurs when small bounding boxes are not correctly trained, resulting in them being misidentified as the backdrop. However, given the model’s size and uses, it is a significant improvement in this use case.



**Fig. 6** Comparative visual representation of the positive image from INRIA dataset which shows difference in terms of precision in the heat map and GradCam image

## 7 Conclusion and Future Scope

The significance of above research lies in deploying pedestrian identification and detection domain in the embedded platforms by using the quantized version of weights and activations in the run time of the deep network. The modules proposed can be used individually, or else the binarized version of the existing neural network can be a game changer to such requirements. Although the research shows of the degradation of the accuracy, using these concepts will improve the real computation time up to 58 times, reduces the size up to 32 time in ideal case, and improves the battery longevity in the portable devices having limited memory. The application of binarized DenseNet to a pedestrian detection system with low latency and processing capacity is discussed in this work. The dense layer of the architecture employs binary weights and activation values, resulting in a significant reduction in precision from  $2^{32}$  to merely 2. This reduces the number of features that it can accommodate inside the network. As a result, under noisy backdrops or with low visibility, the model will be unable to detect the object. To boost the detection recall value, some more layers can be added to the model, which will increase the feature-carrying capacity and reduce data loss due to quantization. Further work should be intended toward how to improve the overall accuracy of such networks and how to make comparable of various neural networks after training with diverse datasets. Some improvements can be done using MeliusNet architecture [17]. In MeliusNet, each dense block is connected to a Melius block, which may improve the results. The feature maps are concatenated to the next layer by shortcut connections, to maximize the feature carrying capacity. Rather than adding new channels directly, the current feature maps are convolutioned with the new channels to increase the

quality of the new channels. Each MeliusNet unit is made up by the dense block and the improvement block, followed by the transition layers. Variants like MeliusNet A and MeliusNet B perform somewhat better than the previous binary DenseNet with reduced computing cost, according to the studies reported in this work [17]. The tests were carried out using the ImageNet dataset, which is a multiclass classification dataset, and the accuracies were observed to be 1–2% higher. Therefore, MeliusNet can provide some additional space for further improving the pedestrian detection in assisted or portable devices.

## References

1. Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
2. Geiger L, Team P (2020) Larq: an open-source library for training binarized neural networks. *J Open Sour Softw* 5(45):1746
3. Tomè D, Monti F, Baroffio L, Bondi L, Tagliasacchi M, Tubaro S (2016) Deep convolutional neural networks for pedestrian detection. *Signal Process: Image Commun* 47:482–489
4. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25:1097–1105
5. Huang X, Ge Z, Jie Z, Yoshie O (2020) NMS by representative region: towards crowded pedestrian detection by proposal pairing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10750–10759
6. Liu W, Liao S, Ren W, Hu W, Yu Y (2019) High-level semantic feature detection: a new perspective for pedestrian detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5187–5196
7. Ojeda FC, Bisulco A, Kepple D, Isler V, Lee DD (2020) On-device event filtering with binary neural networks for pedestrian detection using neuromorphic vision sensors. In: 2020 IEEE International conference on image processing (ICIP). IEEE, pp 3084–3088
8. Courbariaux M, Hubara I, Soudry D, El-Yaniv R, Bengio Y (2016) Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or -1. arXiv preprint [arXiv:1602.02830](https://arxiv.org/abs/1602.02830)
9. Courbariaux M, Bengio Y, David J-P (2015) Binaryconnect: training deep neural networks with binary weights during propagations. *Adv Neural Inf Process Syst*, pp 3123–3131
10. Alizadeh M, Fernández-Marqués J, Lane ND, Gal Y (2018) An empirical study of binary neural networks optimisation. In: International conference on learning representations
11. Bethge J, Yang H, Bornstein M, Meinel C (2019) Back to simplicity: How to train accurate bnnns from scratch? arXiv preprint [arXiv:1906.08637](https://arxiv.org/abs/1906.08637)
12. Rastegari M, Ordonez V, Redmon J, Farhadi A (2016) Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. Springer, Cham, pp 525–542
13. Hubara I, Courbariaux M, Soudry D, El-Yaniv R, Bengio Y (2016) Binarized neural networks. *Adv Neural Inf Process Syst* 29. <https://doi.org/10.48550/arXiv.1602.02830>
14. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
15. Saleh K, Hossny M, Nahavandi S (2019) Real-time intent prediction of pedestrians for autonomous ground vehicles via spatio-temporal densenet. In: 2019 International conference on robotics and automation (ICRA). IEEE, pp 9704–9710
16. Murthy CB, Hashmi MF, Keskar AG (2021) Optimized MobileNet+ SSD: a real-time pedestrian detection on a low-end edge device. *Int J Multim Inf Retrieval* 10(3):171–184

17. Bethge J, Bartz C, Yang H, Chen Y, Meinel C (2021) MeliusNet: an improved network architecture for binary neural networks. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), 2021, pp 1439–1448

# A Mind-Driven Artificial Limb Movement Framework Using Long Short-Term Memory Algorithm



Ahona Ghosh and Sriparna Saha

**Abstract** A rehabilitation framework based on brain-computer interfacing with the help of an electroencephalogram and an OpenManipulatorX robotic arm for people having difficulty in certain body movements has been proposed in this paper. Electroencephalogram signal preprocessing has been done using Butterworth filter and Power Spectral Density has been applied as the feature extractor. Furthermore, long short-term memory has been applied as the motor imagery-based movement classifier and error-related potential detector while designing the mind-driven artificial limb movement framework. The classification performance has been tested based on accuracy, precision, recall, F1-score, and average error rate. Finally, the proposed LSTM showing 94% accuracy in classifying motor imagery action is employed in the robotic arm movement control framework and has shown 56.89 and 63.76% accuracy respectively in real-time imagery detection and error detection. The proposed model outperformed the existing related works after comparing them.

**Keywords** Artificial limb · Electroencephalogram · Long short-term memory · Motor imagery · Motor rehabilitation · OpenManipulatorX

## 1 Introduction

Around 5 million people in India suffer from different disorders in movement/motor functions. People with neuromuscular disorders like Myasthenia gravis, cerebral palsy, multiple sclerosis, etc. are not able to express themselves [1]. Automatic control of prosthetic devices attached to human body limbs bypassing neural communication and motor control is the main benefit of brain-computer interfacing (BCI) based rehabilitation systems. Among different non-invasive and invasive tools of motor imagery-based BCI applications, the electroencephalogram (EEG) has been widely

---

A. Ghosh (✉) · S. Saha

Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal 741249, India  
e-mail: [ahonaghosh95@gmail.com](mailto:ahonaghosh95@gmail.com)

recognized as an effective tool for its cost-effectiveness, portability, and temporal resolution [2].

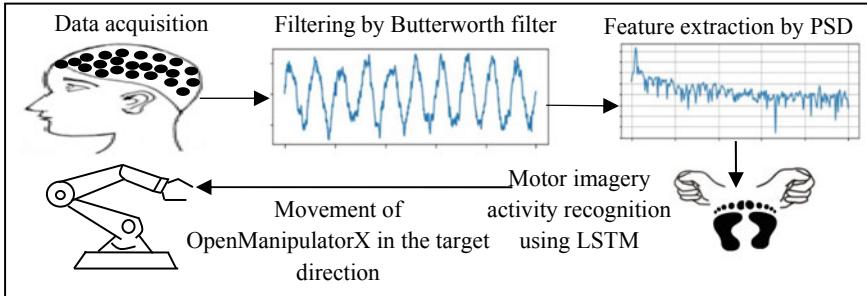
The design and testing of prosthetic limbs and operational electric stimulation algorithms are expensive and risky [3]. Reach and grab is a vital activity in our daily life, and a robotic arm can help to achieve the same for a subject with an arm injury or disorder wishing to bring glass of water but is unable to do so due to disability in arms. This is the major motivation behind current work where accurate classification of motor imagery signal-based robotic arm control has been proposed. Recent studies have shown that human EEG signals can be used to decode object grab types [4], grasp time [5], hand postures, and reach characteristics. For online control of brain-machine interfaces, movement-related spectrum modulation of EEG can be used [6], for example, elbow movement, or for three-dimensional cursor control. Among the various brain signals, event-related synchronization/ desynchronization (ERS/ERD), error related potential (ErrP), steady state visually evoked potential, P300, and slow cortical potential are the most popular [7].

Ferrez et al. [8] tracked EEG's ERS/ERD signal for a discrete set of movements and stopped the tracking on detection of ErrP which led to the cancelation of the last step of the movements. But the proposed position control framework considers the continuous movement of all the links of the robot. Simultaneous tracking of ERS/ ERD and ErrP signal helps to set a positional offset correction as the stopping criteria. Moreover, based on the motor imagery and ErrP detector's satisfactory performance, the system has been validated on a real-time robot and can be concluded as suitable for real-time rehabilitative applications. The current paper proposes an automatic seamless artificial limb control framework using the motor imagery signals collected by EEG and the corresponding movement of a cost-effective and programmable robotic arm OpenManipulatorX. After the filtering and the feature extraction using Power Spectral Density (PSD) [9], to deal with the sequential EEG data, the movement classification is performed by the long short-term memory (LSTM) model [10].

The proposed methodology is explained in Sect. 2. The experimental result and its performance assessment are provided in Sect. 3. Finally, the concluding remarks and possible future scope have been stated in Sect. 4.

## 2 Proposed Methodology

Figure 1 demonstrates the step-by-step working of our mind-driven artificial limb control system for motor rehabilitation.



**Fig. 1** Illustration of the proposed EEG-driven artificial limb movement framework

## 2.1 Introduction to EEG

EEG is a sensor that collects electrical activity within the human brain by attaching electrodes to the subject's scalp [11]. The experiment followed the 10/20 electrode placement technique, where the data has been collected in the proposed framework from nine electrodes, namely F3, C3, P3, FP1, F7, T3, T5, O1, A1.

## 2.2 Filtering Using Butterworth Filter

To accurately estimate the transfer function required in filter creation, we employ several approximation functions in linear analog [12]. Low order Butterworth filter design for its smooth and monotonically decreasing frequency response has been used here for the raw EEG data having a 1 Hz lower cut-off frequency and a 20 Hz upper cut-off frequency. An  $n$ th order Butterworth filter's frequency response is denoted by Eq. (1).

$$H_{j\omega} = \frac{1}{\sqrt{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p}\right)^{2n}}} \quad (1)$$

where,  $\omega$  is equal to 2 and  $\varepsilon$  is the highest passband gain, and  $n$  denotes the filter order ( $A_{max}$ ). Since  $\varepsilon$  is equal to one if  $A_{max}$  is defined at a frequency equal to the cut-off – 3 dB corner point ( $f_c$ ),  $\varepsilon^2$  is likewise one. But if  $A_{max}$  is defined at a different voltage gain value, like 1 dB or 1.1220 (1 dB =  $20 \times \log A_{max}$ ), the new value of  $\varepsilon$  can be obtained by Eq. (2).

$$H_1 = \frac{H_0}{\sqrt{1 + \epsilon^2}} \quad (2)$$

where,  $H_0$  and  $H_1$  are the maximum and the minimum passband gain. Filter's frequency response is quantitatively described by its transfer function using Eq. (3).

$$H_{j\omega} = \frac{V_{\text{out}}(j\omega)}{V_{\text{in}}(j\omega)} \quad (3)$$

where,  $V_{\text{out}}$  represents the output signal voltage,  $V_{\text{in}}$  represents the input signal voltage,  $j$  denotes the square root of  $-1$  ( $\sqrt{-1}$ ), and  $\omega$  is the radian frequency ( $2\pi f$ ).

### 2.3 Feature Extraction Using Power Spectral Density

The frequency content of the signal, or how the signal power is distributed over frequency, can be seen in the power spectral density (power spectrum) [13]. Let  $D_i^t = [d_{i,1}^t, d_{i,2}^t, \dots, d_{i,k}^t, \dots, d_{i,m}^t]$  be the data matrix of filtered EEG signals of dimension  $n \times m$  where  $d_{i,k}^t$  represents the column vector containing  $n$  samples of the amplitudes of EEG belonging to the  $k$ th channel,  $k \in [1, m]$ , and  $t$ th trial for class  $i$ . Considering the input signal as  $X(t)$  and an autocorrelation function denoted by  $R_X(T)$ , the Fourier transform of  $R_X(t)$  represented by  $S_X(f)$  is how we define the Power Spectral Density of  $X(t)$ . The PSD of  $X(t)$  in more detail is given in Eq. (4).

$$S_X(f) = F\{R_X(T)\} = \int_{-\infty}^{\infty} R_X(T) e^{-jr2\pi f} dr \quad (4)$$

where,  $j = \sqrt{-1}$ . From this characterization,  $R_X(T)$  is calculated by performing the inverse Fourier transform of  $S_X(f)$ , which implies in Eq. (5).

$$R_X(T) = F^{-1}\{S_X(f)\} = \int_{-\infty}^{\infty} S_X(f) e^{jr2\pi f} df \quad (5)$$

### 2.4 Classification Using Long Short-Term Memory

Since LSTM deals with the long-term dependency problem faced by the Recurrent Neural Network (RNN) [10], thus it has been preferred over other deep learning architectures in our proposed model. The LSTM cells retain data, and the gates do the memory management using Eq. (6).

$$\tilde{c}^{<t>} = \tanh(w_c [a^{<t-1>}, x^{<t>}] + b_c) \quad (6)$$

where,  $a^{<t>}$  denotes the activation function at time  $(t - 1)$ ,  $w_c$  and  $b_c$  are the weight and bias of  $c$ th memory cell, respectively.

### 2.4.1 Forget Gate

For the proposed work, let the feature matrix of PSD features belonging to  $i$ th class and  $t$ th trial be  $S'_i = s'_{i,1}, s'_{i,2}, \dots, s'_{i,m}$  where  $m$  is the number of channels of the EEG signal. The forget gate can be denoted by Eq. (7).

$$\Gamma_f = \sigma(w_f [c^{<t-1>}, s^t] + b_f) \quad (7)$$

where, sigmoid ( $\sigma$ ) is the activation function associated with forget gate, and the forget weight and forget bias are denoted by  $w_f$  and  $b_f$ , respectively.

### 2.4.2 Input Gate

The input gate adds useful information to the cell state represented by Eq. (8).

$$\Gamma_u = \sigma(w_u [c^{<t-1>}, s^t] + b_u) \quad (8)$$

where, sigmoid ( $\sigma$ ) represents the input activation function and  $w_u$  and  $b_u$  represent the updated weight and bias, respectively. The update gate shown in Fig. 2 refers to the combination of the input and the forget gate.

### 2.4.3 Output Gate

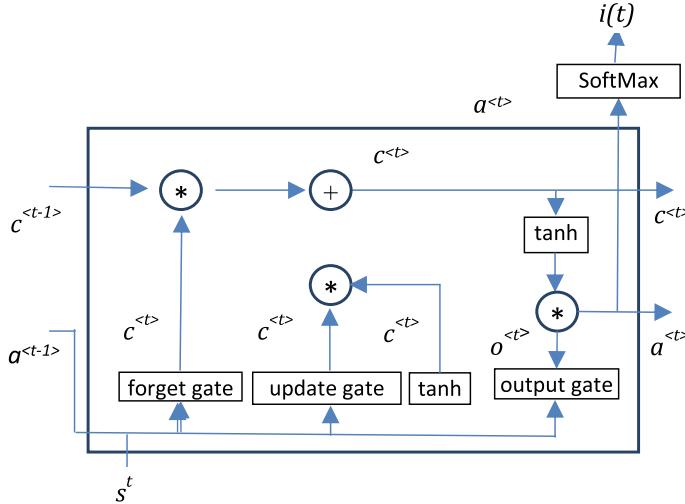
The task of extracting useful information from the current cell state to be presented as output is done by the output gate. It can be denoted by Eq. (9).

$$\Gamma_o = \sigma(w_o [c^{<t-1>}, s^t] + b_o) \quad (9)$$

where, Hthe output weight and output bias are denoted by  $w_o$  and  $b_o$ , respectively. Combining all the outputs, the final memory cell value at time  $t$  is denoted by Eqs. (10) and (11).

$$c^{(t)} = \Gamma_u \times \Gamma \tilde{c}^{(t)} + \Gamma_f \times \tilde{c}^{(t-1)} \quad (10)$$

$$a^{(t)} = c^{(t)} \quad (11)$$



**Fig. 2** Architecture of long short-term memory map

The proposed LSTM architecture is shown in Fig. 2 and the pseudocode of the proposed filtering, feature extraction along with classification process is presented in Procedure 1.

---

**Procedure 1:** Butterworth-PSD-LSTM-based identification of MI signal from EEG data

---

**Input:** Raw EEG signal vector  $R$  consisting of data points like  $r_{i,2}^t, \dots, r_{i,k}^t, \dots, r_{i,m}^t$

---

**Output:** The predicted class label for each unknown data point

**Begin**

**Step 1:** Calculate the fourth-order Butterworth filter using (1), (2) and (3)

**Step 2:** Obtain the filtered data point vector  $D_i^t = [d_{i,1}^t, d_{i,2}^t, \dots, d_{i,k}^t, \dots, d_{i,m}^t]$

**Step 3:** Calculate the PSD of signal  $X(t)$  using (4)

**Step 4:** Obtain the feature matrix belonging to  $i$ th class and  $t$ th trial  $S_i^t = s_{i,1}^t, s_{i,2}^t, \dots, s_{i,m}^t$

**Step 5:** Calculate previous cell output  $\tilde{c}^{<t-1>}$  using (6)

**Step 6:** Feed  $S_i^t$  and  $\tilde{c}^{<t-1>}$  to the forget gate

**Step 7:** Obtain binary input by (7) from  $\sigma$  activation function

**Step 8:** Update the gate using (8)

**Step 9:** Apply output gate (9) to the output of update gate

**Step 10:** Obtain the final memory cell value at time  $t$  using (10) and (11)

---

**End**

---



**Fig. 3** Experimental setup in MAKAUT, WB laboratory

### 3 Experimental Results and Discussion

The experimental framework consists of a desktop with 16 GB RAM, an Intel i5 processor, Ubuntu 16.04, Python 5.2 installed in it, along with a 21-channel EEG and an OpenManipulatorX robotic arm.

#### 3.1 *Experimental Framework*

Figure 3 shows the experimental setup in our MAKAUT, WB laboratory, where 6 healthy subjects (four females and two males with age range 23–33) participated in the experiment. Channels got selected based on the associated regions of the brain to perform activities according to the commands. All the subjects are asked to imagine turning left, right, moving forward, and moving backward in a sequence. The sampling rate of EEG has been set as 512 Hz. The duration of the data in each trial from each subject is approximately 10 s for each activity. The execution of four activities each for 35 trials led to the data size of 15 channels  $\times$  6 subjects  $\times$  20,480 frames  $\times$  35 trial that is split into training and testing data in a 7:3 ratio.

#### 3.2 *Performance Assessment*

True positive rate (TPR), false positive rate (FPR), accuracy, precision, recall, F1-score, and average error rate (AER) have been considered to estimate the projected framework's applicability.

**Table 1** Comparative evaluation of the proposed feature extraction approach

Feature extraction approach	TPR	FPR	Accuracy
Common spatial pattern	0.78	0.16	0.65
Features from the time domain	0.72	0.34	0.74
Features from the frequency domain	0.66	0.32	0.52
Power spectral density (proposed)	0.89	0.08	0.94

### 3.2.1 Feature Extraction

The proposed Power Spectral Density approach has been compared with some existing feature extraction methods, namely Common Spatial Patterns, features of respective time, and frequency domains applied to the same dataset based on three widely known performance metrics, i.e., TPR, FPR, and classification accuracy. However, it is clear from Table 1 that the suggested PSD is the best suitable feature extractor for the proposed framework.

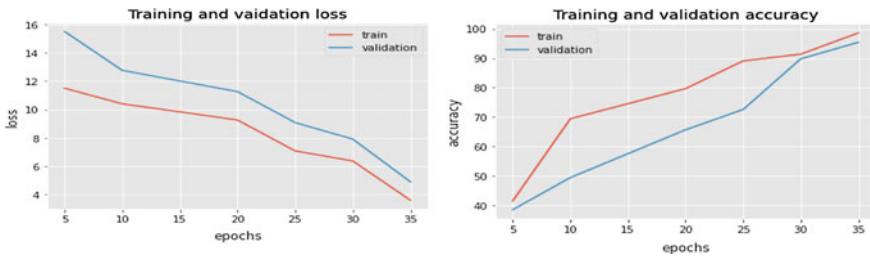
### 3.2.2 Classification

Due to the sequential nature of time series data of EEG, LSTM has been employed as the classifier in our framework. Right, left hand, and right and left foot motor imagery has been classified using hyper-tuned LSTM. To achieve the best performance, the proposed number of LSTM layers, number of hidden units, and state activation function are set as 100, 64, and tanh, respectively, and the learning rate is considered as 0.01. Table 2 demonstrates the accuracies obtained from different combinations of gate activation function, dropout rate, and optimization function where the best accuracy has been marked with bold.

The loss and accuracy of the training and validation phase respectively for 35 epochs are presented in Fig. 4. In the accuracy graph, initial validation accuracy is below 0.45 but with each epoch, it gradually increases upto 0.95. Similarly, the initial

**Table 2** Hyperparameter tuning

	Gate activation function	Optimization function	Dropout rate	Accuracy
Sigmoid	<b>Sigmoid</b>	<b>Adam</b>	<b>0.2</b>	<b>0.94</b>
			0.1	0.91
	Gradient descent		0.2	0.93
			0.1	0.89
SoftMax	SoftMax	Adam	0.2	0.72
			0.1	0.71
	Gradient descent		0.2	0.88
			0.1	0.76



**Fig. 4** Accuracy and loss graph for training and validation phase

**Table 3** Comparative analysis of classifiers used in the existing literature

References	Accuracy	Precision	Recall	F1-score	Avg. error rate
[14]	0.76	0.87	0.34	0.56	0.23
[15]	0.82	0.78	0.73	0.83	0.17
[16]	0.44	0.53	0.48	0.59	0.12
Proposed	0.94	0.92	0.78	0.87	0.06

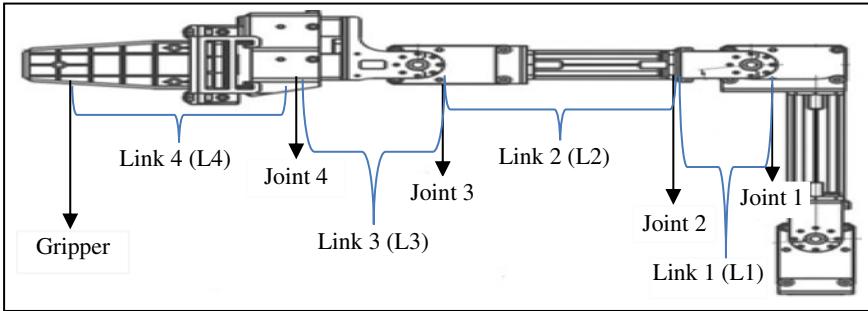
validation loss is above 0.16 but after one epoch decreases upto 0.04. Thus, there is a positive trend toward improving accuracy and reducing loss. At first, validation accuracy is low, but it progressively improves to nearly 97.5%.

### 3.3 Comparison with Related Works

The proposed classifier used in the design of a mind-driven artificial limb has been compared with some existing classifiers of the latest literature based on accuracy, precision, recall, and average error rate. The methods introduced in the related works have been applied to the dataset considered in the present work. Table 3 shows a comparative analysis with [14–16] proving the proposed prototype to be the most effective one in its concerned domain.

### 3.4 Real-Time Implementation of the Proposed Framework

The outstanding performance of the classifier in detecting motor imagery action motivated us to implement the system in a real-time scenario. OpenManipulatorX robotic arm has been used here as an artificial limb, which has 5 degrees of freedom, including the gripper to grip the object. The desired links of the robot with their corresponding motor imagery actions considered in our framework have been shown



**Fig. 5** Joints and links of OpenManipulatorX

**Table 4** Encoding motor imagery actions into rotational/translational movements

Motor imagery	Robot actions	Desired link movements
Left hand	Rotation toward left	Clockwise move all the links
Right hand	Rotation toward right	Anti-clockwise move all the links
Left foot	Forward movement	Translate forward the second link
Right foot	Stop	Stop rotations/translations of all the links

in Fig. 5 and Table 4, respectively. For this proposed work, the links have been controlled in ascending order of their numbers.

The performances of the motor imagery detector averaged over L1, L2, L3, and L4 concerning 6 subjects with subject id (S.ID) of 1 to 6 is shown in Table 5 in terms of TPR, FPR, accuracy, and time consumption. It shows that all the subjects in real-time achieve an accuracy of at least 45% having subject 3 performing the best, marked with bold. Also, the classifier's computational time is appropriate for its concerned scenario. The performance of the LSTM-based positional error detector to detect the occurrence of error has also been tested for all the subjects participating in the experiment, and for all of them, the accuracy, TPR, FPR, and time consumed to detect the error have been presented in Table 6. Subject 6 has achieved the best performance with 63.76% accuracy, 0.91 TPR, 0.11 FPR, and 0.164 s computational time.

**Table 5** Motor imagery recognition performance of OpenManipulatorx

S. ID	Accuracy (%)	TPR	FPR	Time (s)
1	53.33	0.64	0.33	0.264
2	51.35	0.59	0.45	0.231
3	45.67	0.49	0.49	0.324
<b>4</b>	<b>56.89</b>	<b>0.71</b>	<b>0.29</b>	<b>0.237</b>
5	48.21	0.54	0.41	0.293
6	54.12	0.59	0.37	0.275

**Table 6** Performance of positional error detector of OpenManipulatorx

S. ID	Accuracy (%)	TPR	FPR	Time (s)
1	54.33	0.77	0.31	0.216
2	61.25	0.56	0.21	0.284
3	58.35	0.43	0.56	0.264
4	61.31	0.63	0.41	0.236
5	59.64	0.69	0.48	0.121
<b>6</b>	<b>63.76</b>	<b>0.61</b>	<b>0.11</b>	<b>0.164</b>

## 4 Conclusion

The proposed mind-driven artificial limb movement framework is suitable to be applied in real time for patients with upper limb impairments. OpenManipulatorX, one of the latest robotic arms, has proved its efficiency in various real-time rehabilitative platforms. The combination of fourth-order Butterworth filter, power spectral density, and long short-term memory as a filter, feature extractor, and classifier has performed better than the related methods. The accuracy of the motor imagery detector and the error-related potential is 56.89 and 63.76%, respectively, which is satisfactory in real-time scenarios. The proposed scheme is easy to use and includes time delays of 1 s between two subsequent link movements to ensure reliability.

However, the proposed work considers a specific order of link selection whereas practical situations may need random choosing of links and simultaneous control of multiple link movements leading to optimal performance of the system. In near future, these two aspects will be attempted to solve more intricate real-time problems.

## References

1. Livneh H, Antonak RF (1994) Review of research on psychosocial adaptation to neuromuscular disorders: I. Cerebral palsy, muscular dystrophy, and Parkinson's disease. *J Soc Behav Pers* 9(5):201

2. Thakor NV, Tong S (2004) Advances in quantitative electroencephalogram analysis methods. *Annu Rev Biomed Eng* 6:453–495
3. Franzluebbers A, Johnson K (2019) Remote robotic arm teleoperation through virtual reality. In: *Symposium on spatial user interaction*, pp 1–2
4. Fifer MS, Hotson G, Wester BA, McMullen DP, Wang Y, Johannes MS, Katyal KD, Helder JB, Para MP, Vogelstein RJ, Anderson WS (2013) Simultaneous neural control of simple reaching and grasping with the modular prosthetic limb using intracranial EEG. *IEEE Trans Neural Syst Rehabil Eng* 22(3):695–705
5. Hauschild M, Davoodi R, Loeb GE (2007) A virtual reality environment for designing and fitting neural prosthetic limbs. *IEEE Trans Neural Syst Rehabil Eng* 15(1):9–15
6. Palankar M, De Laurentis KJ, Alqasemi R, Veras E, Dubey R, Arbel Y, Donchin E (2009) Control of a 9-DoF wheelchair-mounted robotic arm system using a P300 brain computer interface: Initial experiments. In: *2008 IEEE international conference on robotics and biomimetics*, pp 348–353. IEEE
7. McFarland DJ, Sarnacki WA, Wolpaw JR (2010) Electroencephalographic (EEG) control of three-dimensional movement. *J Neural Eng* 7(3):036007
8. Ferrez PW, Millán JDR (2008) Simultaneous real-time detection of motor imagery and error-related potentials for improved BCI accuracy. In: *Proceedings of the 4th international brain-computer interface workshop and training course*, No. CONF, pp 197–202
9. Paul S, Ghosh A (2023) A study on the application domains of electroencephalogram for the deep learning-based transformative healthcare. *Generative Advers Netw Deep Learn Theory Appl* <https://doi.org/10.1201/9781003203964>
10. Mitra A, Biswas A, Chakraborty K, Ghosh A, Das N, Ghosh N, Ghosh A (2022) A machine learning approach to identify personality traits from social media. In: *Machine learning and deep learning in efficacy improvement of healthcare systems*. CRC Press, pp 31–59
11. Combaz A, Chumerin N, Manyakov NV, Robben A, Suykens JA, Van Hulle MM (2012) Towards the detection of error-related potentials and its integration in the context of a P300 speller brain–computer interface. *Neurocomputing* 80:73–82
12. Selesnick IW, Burrus CS (1998) Generalized digital Butterworth filter design. *IEEE Trans Signal Process* 46(6):1688–1694
13. Elson JM, Bennett JM (1995) Calculation of the power spectral density from surface profile data. *Appl Opt* 34(1):201–208
14. Efe MÖ (2008) Fractional fuzzy adaptive sliding-mode control of a 2-DOF direct-drive robot arm. *IEEE Trans Syst Man Cybern Part B Cybern* 38(6):1561–1570
15. Thallemer AM (2016) AirArm: an anthropofunctional robot arm with inherent flexibility. <https://hdl.handle.net/1783.1/99972>
16. Hoffman G, Ju W (2014) Designing robots with movement in mind. *J Hum-Rob Interact* 3(1):91–122

# Integration of Blockchain Technology with Renewable Energy for Sustainable Development: Issues, Challenges and Future Direction



Saumendra Das, H. Swapnarekha, and S. Vimal

**Abstract** Today, the renewable energy sources such as air, water, geothermal and sun have appeared as an efficient alternative for the smart grid (SG). These renewable energy sources (RES) are considered as prime source of energy for sustainable development to protect the world from climate change. However, few limitations like the political will or good governance have hindered the distribution of these renewal energies sources. Further, the poor demand and consumers' interest in SG are also challenging for sustainable development. To comprehend the difficulties on awareness building and sustainable development of RES, the blockchain technology (BT) could be the key way for the distribution and management of energy resources. Above all, the BT could act as media to integrate the producer and consumer with its flexible and efficient form of distribution or management process. Moreover, the BT preserves the trust, security and confidentiality of keeping or maintaining customer data and also has the opportunity to develop the network. Adoption of RES and decentralized energy distribution with BT integration may improve the viability of renewable energy in the future. This paper discussed on the application and prospects of the BT for the sustainable development of the RES. Again, the entire paper has critically analyzed the pros and cons of integration and future challenges with help of a systematic review of academic literature for a decade.

**Keywords** Sustainable development · Renewal energy technology · Blockchain technology · Decentralized distribution

---

S. Das (✉)

School of Management Studies, GIET University, Gunupur 765022, India  
e-mail: [somu.das2110@gmail.com](mailto:somu.das2110@gmail.com)

H. Swapnarekha

Department of Information Technology, Aditya Institute of Technology and Management, Tekkali, K Kotturu, Srikakulam, Andhra Pradesh 532201, India

S. Vimal

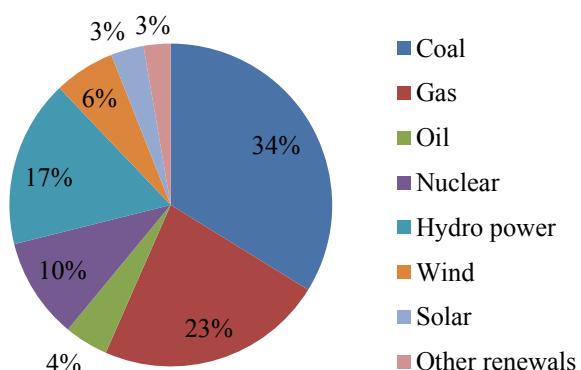
Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu 626117, India

## 1 Introduction

The substantial use of fossils fuel such as coal, petroleum and natural gas for energy sources became hazardous for the environment. The extracts like toxicants, gasoline and methane from various fossil fuel have imbalanced the ecosystem. Particularly, the decomposition of deadly animals has created a carbonized layer in the environment and affected many living organisms. The enormous consumption of non-renewable sources has changed the climatic condition and created an unhealthy condition worldwide. Despite the rapid global development of renewable energy sources, fossil fuels remain the most widely used form of energy. In 2019, the source estimates that gasoline, coal and natural gas accounted for about 80.2% of all energy consumptions worldwide [1]. The need for renewable energy has actually arisen from the increasing demand for energy, which is more concerned with the usability of fossil fuels and their important role in the process of sustainable development. Further, the renewal energy sources are considered as an important source for satisfying the customer demand in many developing countries. However, the benefits and awareness from the renewable energy source (RES) have made unsatisfactory growth in many parts of the world [2]. Consequently, the market shares of renewable energy technologies (RET), government support, public attention, increasing interest of consumers and environmental consequence have tremendously reformed the consumption pattern. According to the source, almost 60% of the world's energy source was generated from the burning fossil fuels like oil, coal and natural gas and 30% world's electrical energy produced from various renewable including solar power, wind energy and hydropower [3]. Figure 1 exhibits the world electricity production (in percentages) from various sources.

Global energy consumption seems to be depending on fossil fuels. Since 50 years, energy consumption significantly increased and undertook a shift in the usage of fossil fuels. Further, it observed that coal energy consumption was recorded 39% in 1965 and substantially declined to 27% in 2020. Over the same period, oil usage is reduced to 40% in 1965 and 31% in 2020. The fossil fuel like natural gas usage has increased

**Fig. 1** World electricity production (in percentages)

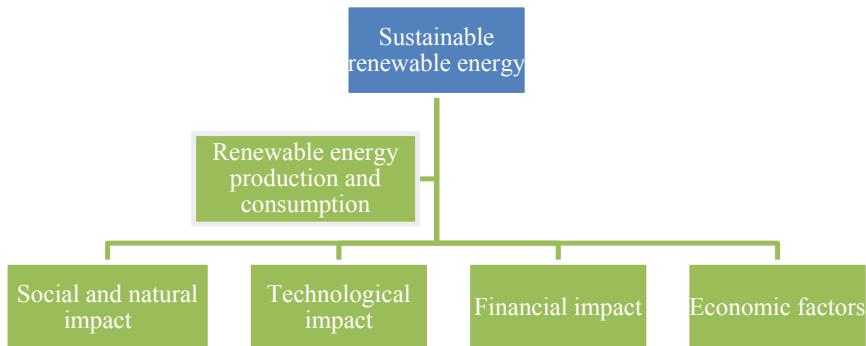


significantly from 15 to 25% in 2020 [4]. However, in recent days, the share of renewable is only 5% among all energy usage. Still, this segment remains insignificant but proved to be sustained and exponential. The widespread adoption of fossil fuels hampered chances for decarbonisation and alternative energy sources. Thereby, the policy makers, government and public should adopt the RET for their individual and social consumption to reduce pollution, conserve the natural resource and provide quality energy with less cost [5]. The renewable energy from various sources such as wind, water, geothermal or bio-fuels strengthens the nation's economy and substitution for conventional source. Moreover, the partial adoption of renewable energy with the help of IoT, big data analytics and cyber system management could improve the performance and productivity of energy industry. In this connection technical and skill development training programs could foster the renewable energy consumption and distribution systematically [6].

To fight against the climate change and global warming, now the policymakers are switching from fossil fuels to renewable energy. Particularly, the involvement of people by decentralized energy distribution in regional areas, adoption of recent technologies and public awareness are the key areas of sustainable energy development. In emerging nations, where the combination of renewable energy and BT expands the energy supply in local markets through a decentralized distribution, the use of blockchain technology (BT) and customer co-ownership is seen as the wise choice for sustainable energy arrangements [7, 8]. Now, BT has gained popularity for its self-recording system like a chain. It provides safe transactions for logistics and supply chain in different areas like health, money transfer, insurance and external trade and also has strong distribution system with suppliers [9]. Therefore, BT is a systematic network enabling system for distribution, pricing and promotion of renewal energy technologies between the consumers and producers extensively. The integration of RET and BT is a noble idea of research where the prosumers and consumers could exchange the renewable energy in a decentralized distribution to accelerate the nation's economy. The sections of this paper are as follows: Sect. 2 is about sustainable renewable energy: an overview; Sect. 3 revealed the advantages and disadvantages of renewable energy; Sect. 4 demonstrated the integration of BT and RE; Sect. 5 deals with the critical analysis; and Sect. 6 articulated the conclusion, suggestions and future direction of the research.

## 2 Sustainable Renewable Energy: An Overview

Future energy requirements in both urban and rural are significantly more. Substantial usage of renewable energy is crucial for achieving the objective of sustainable development. The major justifications for adopting RET are the decentralized distribution system, pollution-free operation and preservation of natural resources. So, the government should conduct surveys to determine the potential market for green energy technologies in both urban and rural regions in order to assist in their promotion [10]. Figure 2 exhibits various considerations on RET.



**Fig. 2** Sustainable growth and development of RET

## 2.1 Social and Natural Impact of Renewal Energy

The rapid rise of RES has created numerous issues for the smart grid (SG) energy system recently. The main challenge of SG is infrastructure facilities and limited resources. To overcome the challenges, RES is considered as an important source of energy across the world. By utilizing cutting-edge technologies like BT, AI and IoT, sustainable development of renewable energy could be improved through appropriate network security. Once more, these technologies will demonstrate energy waste and advance the peer-to-peer network [11]. Recent studies revealed that the reformation of distributed energy and the incorporation of industry 4.0 for corporate sustainability might balance the economy on both the macro and micro levels. Through the use of BT, green energy output will adhere to the long-term preservation of nature [12]. Table 1 exhibits the contributions on social and natural impacts of RE.

**Table 1** Contributions on social and natural impacts of renewable energy

S. no.	Contributions	Recommended outcomes	References
1	Sustainable RE consumption in rural India	Decentralized distribution of RE in rural habitants	[13]
2	Sustainable development by Industry 4.0	Digitalized distribution of RE	[14]
3	Psychological behavior toward RES adoption	Value, belief and norms	[15]
4	Sustainable energy for all	Socio-technical assemblage	[16]
5	Sustainable microloan	SHS for poverty reduction	[17]

**Table 2** Contributions on technological impact of renewable energy

S. no.	Contributions	Recommended outcomes	References
1	Sustainable RE with BT	Peer-to-peer trading	[20]
2	AI application in RES	AI evaluation model for forecasting RE	[21]
3	Sustainable application of BT	BT improve privacy	[22]
4	BT in distributed PV industry	Strengthen the production, distribution	[23]
5	Materialization of BT	Peer-to-peer energy deal	[24]
6	Integration of BT in smart sustainable city	Economic, social and environmental transformation	[25]
7	BT for smart villages	Improvement of rural health and poverty	[26]
8	Innovations in irrigation and agriculture through BT	Solar coin for energy and water trading	[27]

## 2.2 *Technological Impact of Renewable Energy*

The rapid growth in smart cities flourished the use of smart technologies. Despite of significant efforts made on renewable energy through sensing technologies, the smart grid consumption is increasing in the cities. Consequently, the consumers are habituated with conventional energy consumption which became a challenging task for policymakers and government. Thereby, a smart solution-based technological application like IoT and BT is deemed to be futuristic solution to enhance the sustainability of renewable energy in cities. Consumers can use these technologies to detect the energy waste and monitor the consumption. Particularly, these technologies will provide a sustainable cost-effective energy system [18]. The nascent technologies like BT and IoT have got the potential to accelerate the dynamics of supply chain management for green procurement activities, waste management, energy management, packaging and transport. Therefore, the integration of BT and IoT could enhance the productivity and overcome all the challenges on security and transparency [19]. Table 2 exhibits the contributions on technological impact of RE.

## 2.3 *Financial Force of Renewable Energy*

Financial support is a significant parameter to strengthen the renewable energy market. The credit facilities to rural consumers through microfinance could impart high consumption of RET. Significant changes in the use of RE have recently been made, with customers providing solar PV, solar homes and agricultural equipment at discounted rates. Though the government improved the market structure for RE, the

**Table 3** Contributions on financial force of renewable energy

S. no.	Contributions	Recommended outcomes	References
1	Financial programs for sustainable renewable energy	Green certificate, tariff rate and tax incentives	[30]
2	The socio-economic and environmental benefits	Less financing costs for renewable	[31]
3	Financial inclusivity and globalization	Energy efficiency gains to boost carbon emission	[32]
4	Energy entrepreneurship and energy democratization	Minimize the financial obligations	[33]

customers are still disagreeing with the usages. Therefore, the project finance facilities should provide to enhance the stabilized market and increase the awareness. Both the financial growth and renewable energy have marginal impact on environment contamination for CO<sub>2</sub> emission and became challenging for sustainable economic growth [28, 29]. Table 3 exhibits contributions on financial force of RE.

## 2.4 Economic Force of Renewable Energy

The evolution of the universal energy system has both opportunity and challenge for an accepted economy. Thus, the improvement of economic conditions in rural areas and regional impacts might be obtained using employment ratio, supply chain and input–output models [34]. The evolution of renewable energy also has potential impact on job creation and employability leading to development of gross domestic product [35]. Further, it is observed that the economic expansion of a country is significantly related to renewable energy consumption because of its cost-effectiveness [36]. In this regard, a balanced economy includes the use of RES. Table 4 exhibits the contributions on economic forces of RE.

**Table 4** Contributions on economic force of renewable energy

S. no.	Contributions	Recommended outcomes	References
1	Sustainable economy	Green employment in Germany	[37]
2	Smart Energy System approach	Job creation in Europe	[38]
3	Electricity generation cost; pay-back time for greenhouse gas emissions	Major sustainable electricity generation is wind and small hydro	[39]
4	Economic welfare	Improve the part of RE consumption	[40]
5	Economic impact of renewable energy	Installation of RES in rural areas	[41]

**Table 5** Contributions on advantage and disadvantage of RE

S. no.	Contributions	Recommended outcomes	References
1	Disadvantage of renewable energy	Failing to fulfill energy demand	[45]
2	Advantage of RET	Incentive strategies and policies	[46]
3	Sustainable economy	Renewable energy policy	[47]
4	Advantages of renewable energy	Energy safety and job creation	[48]

### 3 Advantages and Disadvantages of Renewable Energy

Recently, the strategic transformation of RET has gained its popularity for its environmental friendly approach. The RET such as wind power, geothermal, solar, hydropower, tidal and hydrogen energies are widely used to maintain a zero-carbon future. However, the marketing and sales of RET are now at infant stage due to lack of awareness or government policies. Therefore, the diffusion of RET is very useful for its high upfront cost, or performance, but it has distinct advantage from secured energy, environmental and social consideration [42]. The RE possesses unique features like standalone or household seized energy mostly used in remote areas [43]. Consequently, it has disadvantages like cost-effectiveness, performance or installation of marine energy, tidal power, geothermal, wind energy and ocean thermal energy [44]. Table 5 provides the contributions on advantage and disadvantage of RE.

### 4 The Integration of BT and RET

Distributed energy system is widely familiar as perspective approach of sustainable energy. It enables the decentralization of distribution and sales of renewable energy where secure and flexible operation takes place. BT is an emerging concept to ensure security and trust for decentralized distribution of RET [49]. The future of energy distribution considered to be uncertain for which BT could mitigate the problem with help of peer-to-peer distribution [50]. Though BT facilitates secured distribution and reliable connectedness with consumers, it has also feasibility of exploring infrastructure to integrate RET [51]. With an estimation to attain zero emission by 2050, there is a partial transition of energy through de-carbonization, decentralization and digitalization. BT has enough potential to facilitate distributed, peer-to-peer trading with less transaction costs, increased safety through cryptography and prosumer's choice. Therefore, the integration of BT with renewable energy distribution could stimulate the sustainable transformation [52]. Table 6 provides the contributions on integration of BT with RET.

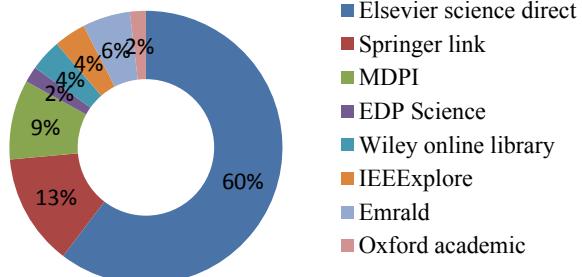
**Table 6** Contributions on integration of BT with RET

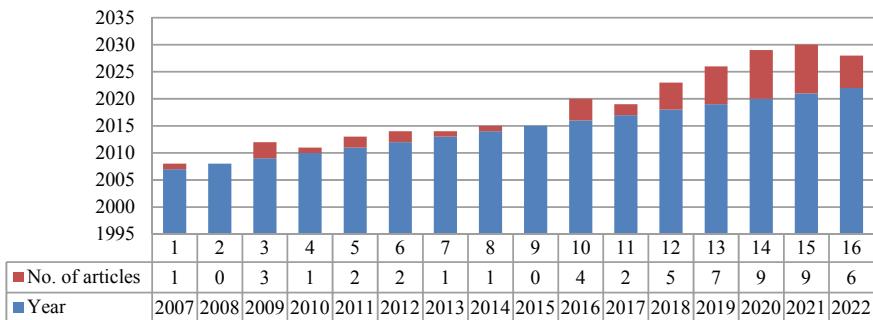
S. no.	Contributions	Recommended outcomes	References
1	Sustainable RE for future	Distributed/decentralized energy for all	[49]
2	Energy sustainability	Integrated BT and RE more focused	[53]
3	Sustainable RE with integration of BT	Circular economy	[54]
4	Sustainable energy supply chains	Faith, trust worthy, transparent and financial exchanges with the supply chain integration	[55]
5	Sustainable supply chain	Blueprint for BT acceptance in future energy systems	[56]

## 5 Critical Analysis

In this section, the data have systematically arranged from various academic literatures for a decade. The entire studies have conducted with a sequence of review from various databases like Science Direct, Web of science, Inform and Wiley online library. The complete study has collected around 53 journal articles, periodicals, book chapter and conference proceedings containing the research inputs on renewable energy, integration of blockchain with renewable energy and sustainable development of RET. In this study, the reviews have classified into three different sections. Section 1 has articulated upon the sustainable development of various renewable energy technologies; Sect. 2 has emphasized the advantage and disadvantage of the renewable energy technology, and finally, Sect. 3 has demonstrated the integration of RET with BT. It is observed that among the various sources of data collection, around 60% of literatures are from Elsevier Science Direct, 13% from the Springer Link, 9% from MDPI publication, 6% from the Emerald Insight, 4% from Wiley Online Library, 4% from IEEE Xplore, 2% from EDP Science and 2% from Oxford Academic. Therefore, it can be said that maximum literatures are from the Elsevier Science Direct. Figure 3 exhibits the distribution articles from various database.

The series of information from various sources are collected over a period of decade. The application of BT with RET has been set up methodically to observe the

**Fig. 3** Sources of literature



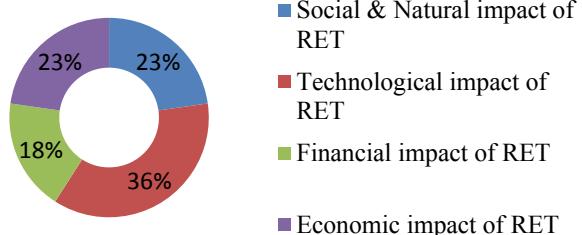
**Fig. 4** Year-wise data of articles

sustainable renewable energy system since 2007. The data revealed that more than nine papers are gathered in the year 2020 and 2021. Figure 4 exhibited the collection of academic literatures.

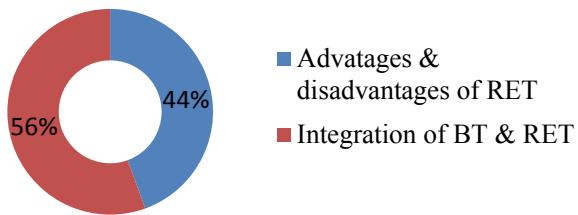
Sustainable development of renewable technology could be implemented through social and natural environment, technological environment, financial and economic development. The data collected from the various sources revealed that 36% of the authors gave their opinion on technological impact of RET, 23% believed that social, natural and economic impact of RET is identified, whereas 18% suggested that financial impact also exists. Therefore, it could be analyzed that technology-enabled services such as BT, IoT, AI have higher degree of concentration on sustainable renewable energy. Figure 5 explores the classification sustainable renewable energy.

The data analysis also observed the important role of technology integration with RET and its advantages and disadvantages. Most of the article revealed on sustainable integration of BT, sustainable supply chain and decentralized distribution network. Thereby, the sustainable development of RET will improve the energy efficiency. In this study, it observed that 56% of articles articulated on the integration technology with RET and 44% emphasized on the advantages and disadvantages of RET. Figure 6 explores on the sustainable RET and its advantages.

**Fig. 5** Classification of sustainable energy system



**Fig. 6** Sustainable RET and its advantages



## 6 Conclusion, Suggestions and Future Direction of Research

In this digital revolution, the renewable energy transition is becoming popular for its operational efficiency and decentralized distribution. The technological adoption is also accelerating the economic efficiency of RET market. In this conceptual review, it observed that sustainable development of RT has been stronger due to massive adoption of technology like BT, IoT and AI. It also explored that public awareness on RET is an important segment for sustainable development. The recent technology has also expanded the utilization of RET for greater possibilities. With the support of BT, the potential consumption of RET is now anticipated to increase more quickly. The scope of marketing, sales and distribution of RET equipment will be expanded in various ways as a result of the integration of BT with RET. The implementation of BT with internet will disperse the RET in rural places, according to this systematic review. Particularly, the prosumers will develop the market potential and improve distribution in local areas. It also suggested that BT will provide very cost-effective renewable energy for all. Therefore, it is crucial to create a strict policy for the implementation of renewable energy technology. Again, the idea was put up that communication barriers would be eliminated via peer-to-peer sharing. Not only the concept of decentralized distribution will promote the renewable technology, but it has other scope like conservation, recycles and balanced economy. Therefore, the study is not limited to application of BT but has significant scope for application of robotics, machine learning or data mining. We cannot restrict the use of renewable energy for domestic or agricultural use, but it must be operationally effective for commercial gain.

## References

1. <https://www.c2es.org/content/renewable-energy/>. Accessed on 25 Jan 2023
2. Sharifi M et al (2019) Forecasting of advertising effectiveness for renewable energy technologies: a neural network analysis. *Technol Forecast Soc Change* 143:154–161. <https://doi.org/10.1016/j.techfore.2019.04.009>
3. <https://www.iea.org/reports/world-energy-balances-overview/world>. Accessed on 25 Jan 2023
4. <https://www.aljazeera.com/news/2022/1/20/interactive-how-much-of-your-countrys-electricity-is-renewable-infographic>. Accessed on 25 Jan 2023

5. Das S et al (2022) Solar photo voltaic renewal energy: analyzing the effectiveness of marketing mix strategies. In: Innovation in electrical power engineering, communication, and computing technology. Springer, Singapore, pp 527–540. [https://doi.org/10.1007/978-981-16-7076-3\\_45](https://doi.org/10.1007/978-981-16-7076-3_45)
6. Arcelay I et al (2021) Definition of the future skills needs of job profiles in the renewable energy sector. *Energies* 14(9): 2609. <https://doi.org/10.3390/en14092609>
7. Fortkort M, Finke S, Severengiz S (2021) Blockchain-based consumer stock ownership plans (CSOP) as a catalyst for impact investments in sustainable energy infrastructure. *Renew Energy Environ Sustain* 6:43. <https://doi.org/10.1051/rees/2021043>
8. Mengelkamp E et al (2018) A blockchain-based smart grid: towards sustainable local energy markets. *Comput Sci Res Dev* 33(1):207–214. <https://doi.org/10.1007/s00450-017-0360-9>
9. Liu J et al (2021) Selection of renewable energy alternatives for green blockchain investments: a hybrid IT2-based fuzzy modelling. *Arch Comput Method Eng* 28(5):3687–3701. <https://doi.org/10.1007/s11831-020-09521-2>
10. Ibrahim D, Rosen MA (2021) Chapter 4—Exergy, environment, and sustainable development, pp 61–89. <https://doi.org/10.1016/B978-0-12-824372-5.00004-X>
11. Kumari A et al (2020) Blockchain and AI amalgamation for energy cloud management: Challenges, solutions, and future directions. *J Parallel Distrib Comput* 143:148–166. <https://doi.org/10.1016/j.jpdc.2020.05.004>
12. Parmentola A et al (2022) Is blockchain able to enhance environmental sustainability? A systematic review and research agenda from the perspective of sustainable development goals (SDGs). *Bus Strat Environ* 31(1):194–217. <https://doi.org/10.1002/bse.2882>
13. Yadav P, Davies PJ, Khan S (2020) Breaking into the photovoltaic energy transition for rural and remote communities: challenging the impact of awareness norms and subsidy schemes. *Clean Technol Environ Policy* 22(4):817–834. <https://doi.org/10.1007/s10098-020-01823-0>
14. Borowski Piotr F (2021) Digitization, digital twins, blockchain, and industry 4.0 as elements of management process in enterprises in the energy sector. *Energies* 14(7):1885. <https://doi.org/10.3390/en14071885>
15. Fornara F et al (2016) Predicting intention to improve household energy efficiency: the role of value-belief-norm theory, normative and informational influence, and specific attitude. *J Environ Psychol* 45:1–10. <https://doi.org/10.1016/j.jenvp.2015.11.001>
16. Kumar A et al (2019) Solar energy for all? Understanding the successes and shortfalls through a critical comparative assessment of Bangladesh, Brazil, India, Mozambique, Sri Lanka and South Africa. *Energy Res Soc Sci* 48:166–176. <https://doi.org/10.1016/j.erss.2018.10.005>
17. Laufer D, Schäfer M (2011) The implementation of solar home systems as a poverty reduction strategy—a case study in Sri Lanka. *Energy Sustain Dev* 15(3):330–336. <https://doi.org/10.1016/j.esd.2011.07.002>
18. Khrais LT (2020) The combination of IoT-sensors in appliances and block-chain technology in smart cities energy solutions. In: 2020 6th international conference on advanced computing and communication systems (ICACCS). IEEE. <https://doi.org/10.1109/ICACCS48705.2020.9074362>
19. Rane SB, Thakker SV (2020) Green procurement process model based on blockchain-IoT integrated architecture for a sustainable business. *Manage Environ Qual Int J* 31:741–763. <https://doi.org/10.1108/MEQ-06-2019-0136>
20. Bao J et al (2020) A survey of blockchain applications in the energy sector. *IEEE Syst J* 15(3):3370–3381. <https://doi.org/10.1109/JSYST.2020.2998791>
21. Chen C et al (2021) Artificial intelligence on economic evaluation of energy efficiency and renewable energy technologies. *Sustain Energy Technol Assess* 47:101358. <https://doi.org/10.1016/j.seta.2021.101358>
22. Teufel B, Sentic A, Barmet M (2019) Blockchain energy: blockchain in future energy systems. *J Electron Sci Technol* 17(4):100011. <https://doi.org/10.1016/j.jnlst.2020.100011>
23. Hou J, Wang H, Liu P (2018) Applying the blockchain technology to promote the development of distributed photovoltaic in China. *Int J Energy Res* 42(6):2050–2069. <https://doi.org/10.1002/er.3984>

24. Thukral, Manish Kumar (2021) Emergence of blockchain-technology application in peer-to-peer electrical-energy trading: a review. *Clean Energy* 5(1):104–123. <https://doi.org/10.1093/ce/zkaa033>
25. Wong PF et al (2020) Potential integration of blockchain technology into smart sustainable city (SSC) developments: a systematic review. *Smart Sustain Built Environ.* <https://doi.org/10.1108/SASBE-09-2020-0140>
26. Kaur P, Parashar A (2021) A systematic literature review of blockchain technology for smart villages. *Arch Comput Method Eng* 1–52. <https://doi.org/10.1007/s11831-021-09659-7>
27. Enescu FM et al (2020) Implementing blockchain technology in irrigation systems that integrate photovoltaic energy generation systems. *Sustainability* 12(4):1540. <https://doi.org/10.3390/su12041540>
28. Charfeddine L, Kahia M (2019) Impact of renewable energy consumption and financial development on CO<sub>2</sub> emissions and economic growth in the MENA region: a panel vector autoregressive (PVAR) analysis. *Renew Energy* 139:198–213. <https://doi.org/10.1016/j.renene.2019.01.010>
29. Hau VB et al (2018) Analyzing the impact of renewable energy incentives and parameter uncertainties on financial feasibility of a campus microgrid. *Energies* 11(9):2446. <https://doi.org/10.3390/en11092446>
30. Abolhosseini S, Heshmati A (2014) The main support mechanisms to finance renewable energy development. *Renew Sustain Energy Rev* 40:876–885. <https://doi.org/10.1016/j.rser.2014.08.013>
31. Sweerts B, Dalla Longa F, van der Zwaan B (2019) Financial de-risking to unlock Africa's renewable energy potential. *Renew Sustain Energy Rev* 102:75–82. <https://doi.org/10.1016/j.rser.2018.11.039>
32. Mursched M et al (2022) The impacts of renewable energy, financial inclusivity, globalization, economic growth, and urbanization on carbon productivity: evidence from net moderation and mediation effects of energy efficiency gains. *Renew Energy* 196:824–838. <https://doi.org/10.1016/j.renene.2022.07.012>
33. Thapar S, Sharma S, Verma A (2016) Economic and environmental effectiveness of renewable energy policy instruments: best practices from India. *Renew Sustain Energy Rev* 66:487–498. <https://doi.org/10.1016/j.rser.2016.08.025>
34. Jenniches S (2018) Assessing the regional economic impacts of renewable energy sources—a literature review. *Renew Sustain Energy Rev* 93:35–51. <https://doi.org/10.1016/j.rser.2018.05.008>
35. Bulavskaya T, Reynès F (2018) Job creation and economic impact of renewable energy in the Netherlands. *Renew Energy* 119:528–538. <https://doi.org/10.1016/j.renene.2017.09.039>
36. Inglesi-Lotz R (2016) The impact of renewable energy consumption to economic growth: a panel data application. *Energy Econ* 53:58–63. <https://doi.org/10.1016/j.eneco.2015.01.003>
37. Lehr U, Lutz C, Edler D (2012) Green jobs? Economic impacts of renewable energy in Germany. *Energy Policy* 47:358–364. <https://doi.org/10.1016/j.enpol.2012.04.076>
38. Connolly D, Lund H, Vad Mathiesen B (2016) Smart energy Europe: the technical and economic impact of one potential 100% renewable energy scenario for the European Union. *Renew Sustain Energy Rev* 60:1634–1653. <https://doi.org/10.1016/j.rser.2016.02.025>
39. Prakash R, Bhat IK (2009) Energy, economics and environmental impacts of renewable energy systems. *Renew Sustain Energy Rev* 13(9):2716–2721
40. Fang Y (2011) Economic welfare impacts from renewable energy consumption: the China experience. *Renew Sustain Energy Rev* 15(9):5120–5128. <https://doi.org/10.1016/j.rser.2011.07.044>
41. Akella AK, Saini RP, Sharma MP (2009) Social, economical and environmental impacts of renewable energy systems. *Renew Energy* 34(2):390–396. <https://doi.org/10.1016/j.renene.2008.05.002>
42. Rao KU, Kishore VVN (2010) A review of technology diffusion models with special reference to renewable energy technologies. *Renew Sustain Energy Rev* 14(3):1070–1078. <https://doi.org/10.1016/j.rser.2009.11.007>

43. Nguyen KQ (2007) Alternatives to grid extension for rural electrification: decentralized renewable energy technologies in Vietnam. *Energy Policy* 35(4):2579–2589. <https://doi.org/10.1016/j.enpol.2006.10.004>
44. Hussain A, Arif SM, Aslam M (2017) Emerging renewable and sustainable energy technologies: state of the art. *Renew Sustain Energy Rev* 71:12–28. <https://doi.org/10.1016/j.rser.2016.12.033>
45. Azarpour A et al (2013) A review on the drawbacks of renewable energy as a promising energy source of the future. *Arab J Sci Eng* 38(2):317–328. <https://doi.org/10.1007/s13369-012-0436-6>
46. Aquila G et al (2017) An overview of incentive policies for the expansion of renewable energy generation in electricity power systems and the Brazilian experience. *Renew Sustain Energy Rev* 70:1090–1098. <https://doi.org/10.1016/j.rser.2016.12.013>
47. Peidong Z et al (2009) Opportunities and challenges for renewable energy policy in China. *Renew Sustain Energy Rev* 13(2):439–449. <https://doi.org/10.1016/j.rser.2007.11.005>
48. Lambert RJ, Silva PP (2012) The challenges of determining the employment effects of renewable energy. *Renew Sustain Energy Rev* 16(7):4667–4674. <https://doi.org/10.1016/j.rser.2012.03.072>
49. Wang Q, Su M (2020) Integrating blockchain technology into the energy sector—from theory of blockchain to research and application of energy blockchain. *Comput Sci Rev* 37:100275. <https://doi.org/10.1016/j.cosrev.2020.100275>
50. Ahl A et al (2019) Review of blockchain-based distributed energy: Implications for institutional development. *Renew Sustain Energy Rev* 107:200–211. <https://doi.org/10.1016/j.rser.2019.03.002>
51. Shojaei A, Wang J, Fenner A (2019) Exploring the feasibility of blockchain technology as an infrastructure for improving built asset sustainability. *Built Environ Project Asset Manage.* <https://doi.org/10.1108/BEPAM-11-2018-0142>
52. Ahl A et al (2020) Exploring blockchain for the energy transition: Opportunities and challenges based on a case study in Japan. *Renew Sustain Energy Rev* 117:109488. <https://doi.org/10.1016/j.rser.2019.109488>
53. Gawusu S et al (2022) Renewable energy sources from the perspective of blockchain integration: from theory to application. *Sustain Energy Technol Assess* 52:102108. <https://doi.org/10.1016/j.seta.2022.102108>
54. Yildizbasi A (2021) Blockchain and renewable energy: Integration challenges in circular economy era. *Renew Energy* 176:183–197. <https://doi.org/10.1016/j.renene.2021.05.053>
55. Almutairi K et al (2022) Blockchain technology application challenges in renewable energy supply chain management. *Environ Sci Pollut Res* 1–18. <https://doi.org/10.1007/s11356-021-18311-7>
56. Juszczyk O, Shahzad K (2022) Blockchain technology for renewable energy: principles, applications and prospects. *Energies* 15(13):4603. <https://doi.org/10.3390/en15134603>

# Binary Classification of Kidney Glomeruli Using Deep Neural Networks



Basra Jehangir, Soumya Ranjan Nayak, and Shaiq Wani

**Abstract** Glomeruli is a collection of blood vessels present in kidneys of the human body. Since kidneys are one of the most important human body organs, diagnosis of any abnormality becomes very important. This work focuses on the binary classification of normal and sclerotic glomeruli using convolutional neural networks from whole slide images (WSI). These images are periodic acid-Schiff (PAS) stained and the glomerulus can be seen as circular areas of dark stains on the slide. The main purpose of this work is to make the diagnoses of the kidney glomeruli fast and accurate since manual detection is quite time-consuming and has many human errors as well. In our work, we have performed the classification of microscopic images to detect the sclerotic glomerulus from the kidney. This study deployed, four different types of CNN models and subsequently evaluated, they are AlexNet, Visual Geometry Group-19, GoogleNet, and a customized model. The comparative analysis has been made by considering several parameters such as the number of epochs, optimizers, batch size, and learning rate in which the customized model achieved the accuracy of 97.86%. The results of the proposed work are quite promising. The performances of the models used in this work are compared using various metrics such as accuracy, recall, precision, and F1-score, and the results from each of the models are noted.

**Keywords** Convolutional neural networks · GoogleNet · AlexNet · Visual geometry group-19 · Whole slide images · Periodic acid-Schiff · Optimizers · Precision · Recall · F1-score

---

B. Jehangir

Amity School of Engineering and Technology, Amity University, Noida 201303, India

S. R. Nayak (✉)

School of Computer Engineering, KIIT Deemed to Be University, Bhubaneswar, Odisha 751024, India

e-mail: [nayak.soumya17@gmail.com](mailto:nayak.soumya17@gmail.com)

S. Wani

Department of Computer Science Engineering, Chandigarh University, Gharaun, Mohali 140413, India

e-mail: [shaiq\\_wani@outlook.in](mailto:shaiq_wani@outlook.in)

## 1 Introduction

Kidneys are one of the most important organs of the human body. They are present in the abdomen behind the ribcage. Kidneys are responsible for the removal of waste. It contains certain structures called the glomeruli. Glomeruli are a collection of blood vessels that are responsible for the removal of waste and unnecessary substances from the body. They are the filters and are responsible for the removal of extra fluids. This makes them a very important part of the kidneys. Damage in any of these glomeruli results in the change of their shape and can eventually lead to chronic kidney disease. In this work, binary classification of kidney glomeruli as normal or cases of sclerosis is performed. For a better understanding of how glomerulus is performed, first an understanding a “nephron” is needed. A nephron is a minute structure in the kidney that is highly functional. It is composed of two parts: renal corpuscle and renal tubule. The former consists of capillaries called the glomerulus. The glomerulus is a cluster of nerve endings, spores, or small blood vessels, it is responsible for filtering the waste from the blood. These glomeruli are constantly filtering the blood in our bodies [1]. Sclerosis of the glomeruli can be caused due to many underlying health conditions like diabetes, obesity, or other kidney diseases. It is usually used for scarring of the kidney glomeruli. A kidney biopsy is necessary for detecting cases of sclerosis. One of the detection methods is using the histopathology slides. Traditionally these slides are observed by medical experts for classification and diagnosis of the disease. However this is quite time consuming, error prone and many times several experts may have a biased opinion about the slides. For elimination of this, we have used an automated diagnosis of the sclerotic kidney glomeruli.

In this work, we have taken the kidney whole slide imaging (WSI) data [2]. The classification dataset available online is used in this work. Further, this data is used for performing the binary classification of the slides and the detection of disease. This is done using convolutional neural networks of four different types GoogleNet, AlexNet, VGG 19, and a customized CNN. By performing comparative analysis, we concluded that out of the four, our model performed the best and the accuracy achieved by the model is 97.86%. To detect and classify the types of glomeruli abnormality in kidneys and the disease associated with it, certain imaging techniques like histopathology slides are used. Further, a comparative analysis of each of the works using this dataset is done for analysis of the work. This work not only provides an optimal solution for fast and accurate classification of kidney glomeruli but also provides a comparative analysis of several convolutional neural networks with respect to certain parameters such as epoch size, datasets, etc.

## 2 Literature Review

Recently various studies have been performed in the field of biomedical image classification and segmentation using various types of convolutional neural networks.

Deep learning has gained huge popularity in microscopy, especially in the analysis of histopathology images and microscopic slides. Disease detection and classification can be determined by processes such as classification and segmentation. Convolutional neural networks have been used in biomedical image classification and segmentation and have shown some promising results. In [3], a detailed study of various techniques for the image segmentation is done along with a thorough comparison, various merits and demerits have been specified in detail. The analysis is based on various parameters which are used to provide a detailed comparison. A detailed description of each layer of a CNN and the function of each layer is provided.

The only factor compromising the performance of a CNN is the size of the dataset, especially in Medicine where data privacy is important. Ronneberger et al. [4] have presented a training strategy that strongly relies on use of data augmentation so that annotated data could be used efficiently. The architecture contains two paths one contracting and the other symmetric expanding path. The author proved that a model that has been given lesser images can outperform other state-of-art models using the International Symposium of Biomedical Imaging (ISBI) challenge of segmentation of neuronal structure. Haque et al. [5] performed an intensive study on the deep learning and their hierarchical feature representation from images, eliminating the need of manual feature extraction and machine learning approaches for biomedical image segmentation. The basics of deep learning models are discussed and various successful implementations are also studied. Various limitations of deep learning for biomedical image processing were addressed like the lack of dataset for training the model, and usage of memory because of storage of the images. Jayapandian et al. [6] used the deep learning U-Net model for segmentation of histopathology images with different stains like silver, periodic acid-Schiff, hematoxylin and eosin, and trichrome into glomerular tuft, glomerular unit, distal tubular segment, proximal tubular segment, peritubular capillaries, and arteries/arterioles. Wang et al. [7] used different encoder neural network architectures such as LeNet-5, AlexNet5, VGG16, InceptionV3, ResNet101, DenseNet201, ResNetXt101, ENet, and Xception for the segmentation of the whole slide images (WSI) and provided potential ways for further improvements. Bueno et al. [8] performed semantic segmentation of periodic acid-Schiff stained whole slide images using U-Net and SegNet into non-glomerulus, normal glomerulus, as well as sclerosis glomerulus for three class classification and binary classification into non-glomerulus and glomerulus using various CNN models. Fu et al. [9] performed image segmentation on a dataset of cardiac histological images with a network that comprised of 11 layers which outperformed the SOTA methods. The proposed method was more efficient and was less susceptible to over-fitting with a dice similarity coefficient of 0.947. Kannan et al. [10] developed a DL framework consisting of a convolutional neural network trained with images as input and labels as output to accurately identify and segment glomeruli from images of kidney biopsies. The overall accuracy of the test data is  $92.67 \pm 2.02\%$  and Kappa is  $0.8681 \pm 0.0392$ . Sun et al. [11] proposed a similarity-based region proposal network (SRPN) to accurately detect the nuclei and cells in histopathology images. Liu et al. [12] proposed a method for nuclei segmentation from pathology images. The proposed methodology uses a deep CNN to perform the segmentation of image slides. The

results of the work are quite promising in comparison with the other state-of-art methods that are already present.

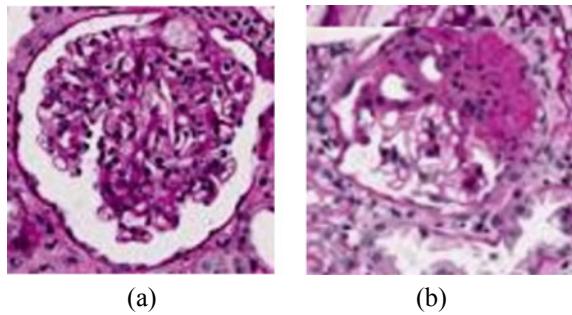
Apart from pathology slides, other images such as computed tomography are also used in disease detection using visual images. Chen et al. [13] performed segmentation of computed tomography (CT) scans using SegNet for the detection of lung cancer, specificity, sensitivity, overlap rate, total image segmentation time, and accuracy. The model achieved a specificity of 86.67%, sensitivity of 98.33%, an accuracy of 92.50%, and a total segmentation time of 30.42 s, which is shorter than manual segmentation. Cruz et al. [14] performed segmentation of kidney CT scan images using KiTs dataset [15] using U-Net 2D and further performed false positive reduction using image processing. Gong et al. [16] performed segmentation and classification of kidney tumors using a 2D ScNet and achieved an accuracy of 99.5% in the binary classification of tumors into benign and malignant. Manjunath et al. [17] performed semantic segmentation using convolutional neural networks and segmented liver computed tomography scans and lesions from the liver segmented part. The model achieved an accuracy of 99.71% for liver and 99.72% for tumor segmentation. Zhao et al. [18] performed segmentation of kidney and kidney tumors via the CT scans of the kidney images using three-dimensional U-Net and MSS U-Net. The method showed superior performance when compared with the state-of-art method and dice coefficient achieved by the method was 0.969 and 0.805, respectively. Anand et al. [19] performed segmentation of skin lesions using a modified U-Net model by considering different parameters such as optimizers, batch size, and epochs and recorded the results.

### 3 Dataset Overview and Preprocessing

The methodology adopted by us is represented in this part of the paper. The entire process from data acquisition to the result analysis has been extensively mentioned.

#### 3.1 Data Acquisition

The dataset that has been taken from open source [1] to detect the sclerosed glomeruli (FTU) [20] from the given dataset. The data that is present contains images in two classes. All the images are in .png format. The total images are 2340 for both normal and sclerosed slides. The source contains two datasets one for segmentation and the other for classification. The data is generated from the detection of glomeruli from images in the former dataset. After the conversion of the images into a smaller size of  $256 \times 256$ , the sample images from either of the classes are visualized in Fig. 1.



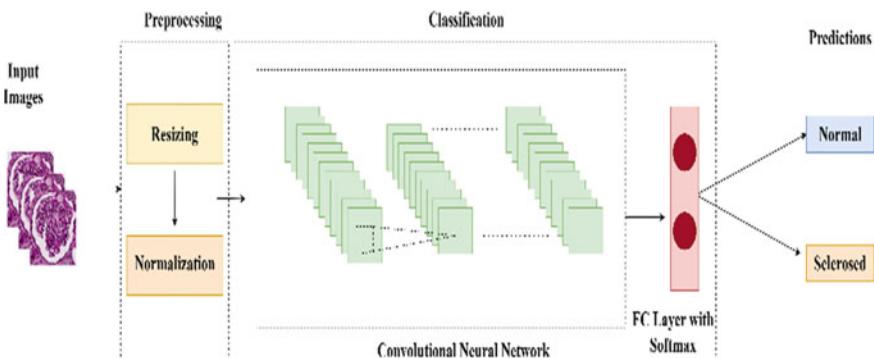
**Fig. 1** Sample images from the dataset: **a** Normal, **b** Sclerosed

### 3.2 Preprocessing

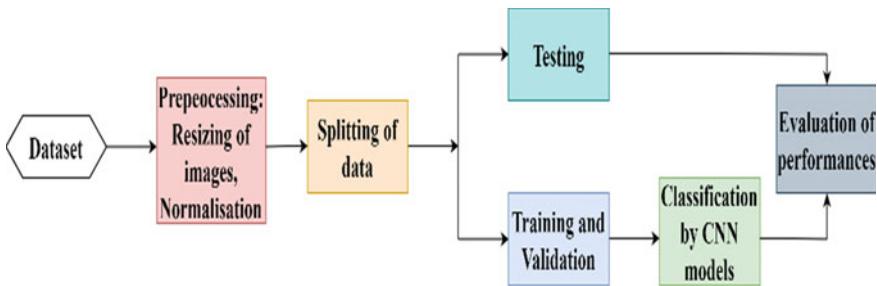
The input scans are of varying sizes. These scans are brought down to the size of (224, 224) in red-green-blue (RGB) format for uniformity. These images are then normalized so that the data can be inserted into the CNN model. The data is further segregated into training, testing, and validation in the ratio of 6:2:2 for different purposes: training data for training the model, validation data for hyper-parameter tuning, and testing data for evaluating the model.

## 4 Methodology Used

Classification may be defined as a process of dividing the data into different groups based on the labels given to us. In recent times, CNN has been gaining huge importance in the classification and segmentation of image data. The overview of the methodology followed in the work is shown in Fig. 2.



**Fig. 2** Overview of the proposed work



**Fig. 3** Overall process followed in the work

Figure 1 clearly illustrates the classification of the dataset using a CNN. The input data is taken and preprocessing is performed which includes resizing and normalization. The data is inserted into the network and after passing through fully connected layer (FCL), prediction is done. The prediction is either normal or sclerosed. This process is utilized in all the models used in the work.

In this paper, several CNN models are used in order to perform binary classification of the images into normal and scleroses glomeruli. The overall process followed in this paper can be seen in Fig. 3.

#### 4.1 The CNN Models

There are four convolutional neural networks used in this paper, namely AlexNet, VGG, Google Net, and customized network. These CNN models were trained over the WSI images and predictions were made. An extensive comparative analysis was made of the results obtained from the models. Figure 3 illustrates the overall pipeline of the work from the very initial step of data acquisition till the evaluation of the performance of models individually. Each step is explained in detail further in the paper.

AlexNet was introduced in order to form a standard for image classification. It contains dropout layers and used ReLU as an activation function. Extensive experimentation proved that even displacing any single layer of this network could change results to a greater extent. VGG 19 or visual geometry group is a convolutional neural network that has 19 layers (3 fully connected layers, 16 convolution layers, 5 maxPool layers, and 1 softmax layer). The architecture of VGG contains blocks and each block contains 2D convolutional layer and max-pooling layers. The network takes a (224, 224, 3) RGB image as the input. The GoogleNet architecture contains 22 layers excluding 5 pooling layers. GoogleNet takes an input of the dimensions  $224 \times 224$ . GoogleNet was designed to improve the efficiency of classification. GoogleNet architecture consists of nine inception modules. There are two max-pooling layers

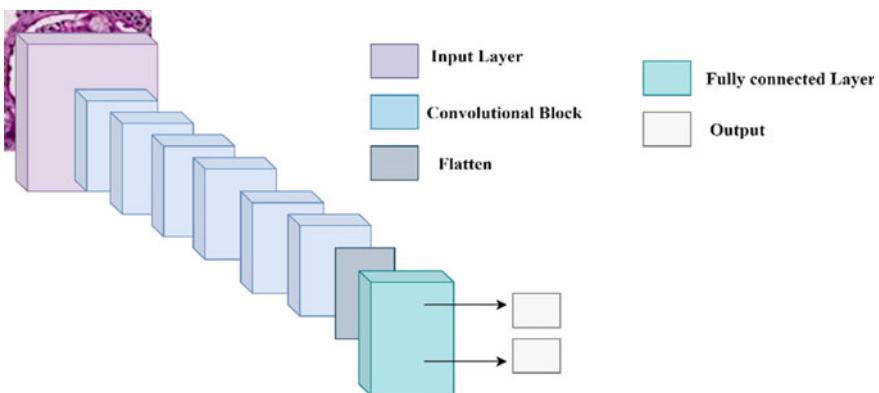
between inception modules. It further contains drop out layer for preventing overfitting of the model. The final layer is the softmax layer which uses the softmax activation function to derive the probability distribution of numbers in the input vector.

## 4.2 Proposed Model

A customized model consisting of CNN layers is used to train on the dataset. The model contains only six convolutional blocks. The rectified linear unit, convolutional layer, and pooling layer are used to extract features from the dataset. The model was made to run over 15 epochs like other models and the results generated were noted. The detailed structure is shown in Fig. 4. Each blue block represents a convolutional block consisting of convolutional 2D, max-pooling layer, and a dropout layer. And the fully connected layer contains two dense layers.

In Fig. 4, the structure of the convolutional neural network can be seen. The conv layer is the primary layer that concentrates highlights from the images. It carries out a numerical product among the input image network and filter. The filter moved bit by bit and carries out image-wise multiplication among the image and the filter. As per the main layer, a filter size of 32 was utilized, which was expanded to 64, 128, 256, 512, and 1024 as the layers continue to increment. The value of the step decides the amount we get the channel across the picture. In this work, the stride is kept at 1. The padding that we utilized for the CNN in this work is the equivalent to filter, so that the filter completely overlaps the incoming scan.

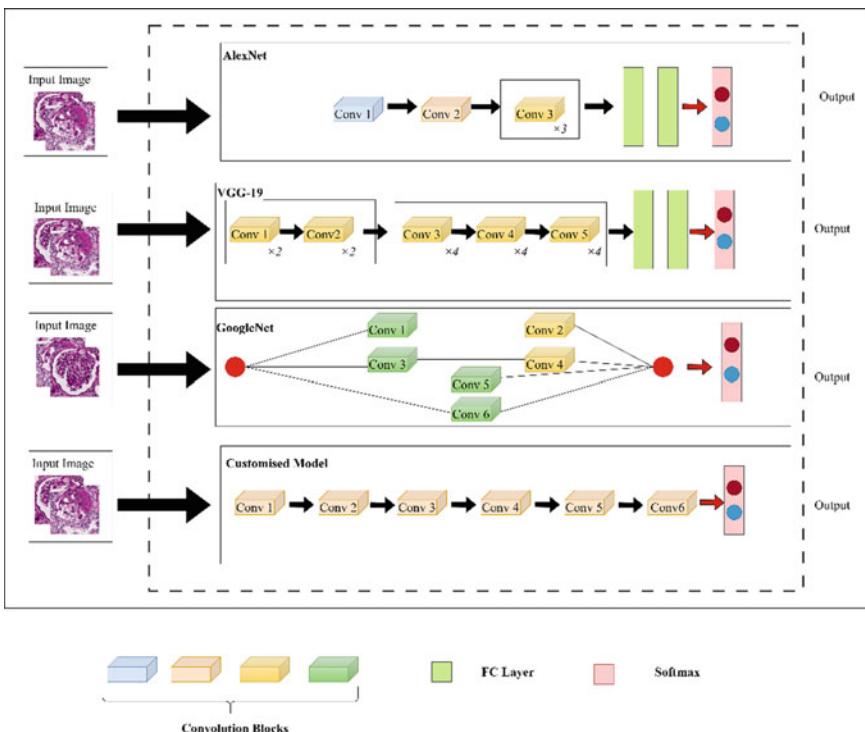
The rectified linear unit sets all the negative values to zero thereby introducing nonlinearity to data. The pooling layer is like the convolutional layer since it lessens the spatial size of the maps. The max-pooling is used in the model so that the overfitting is reduced. Finally, a flattening layer is used to reduce the dimensions of the



**Fig. 4** CNN model architecture used in this work

image matrix and the reduced images are then fed to the FCL with 32 neurons. The activation function used is “sigmoid” since binary classification is performed on the images. The structure is visible in Fig. 4. The model was run over 15 epochs since higher epoch values would cause the accuracy values to saturate. Figure 5 illustrates the major parts of the three convolutional neural networks and our model that has been used for kidney glomeruli detection. The convolutional blocks of different colors show the difference in the size of filters. The prediction of each of the models is clearly visible in the output.

The architectural overview of the trained networks can be seen in Table 1 where AlexNet is 8 layered, VGG-19 is 19 layered, GoogleNet is 22 layered, and the customized model is 6 layered.



**Fig. 5** Overall structure of the proposed work

**Table 1** Architectural description of convolutional neural networks

Model	Layers	Number of parameters (in millions)	Input parameters	Output layer size
AlexNet	8	60	(224, 224, 3)	(2, 1)
VGG-19	19	138	(224, 224, 3)	(2, 1)
GoogleNet	22	5	(224, 224, 3)	(2, 1)
Our model	6	6	(224, 224, 3)	(2, 1)

## 5 Result Analysis

All the models above were run on 15 epochs as the increase in the number of epochs made the accuracy values more saturated. The values of accuracy and loss for each epoch for all four models can be seen in Table 2. The graphical representation of accuracy and the respective epochs is present in Fig. 7.

In the Table 2 we can see the variation of validation loss, train loss, and validation accuracy with the number of epochs. Table 2 illustrated that the accuracy of our model is better than other models used in this work.

**Table 2** Values for loss and accuracy for training and validation

Model	Epoch	Train loss	Valid loss	Valid accuracy
AlexNet	1	0.6923	0.6932	0.5085
	2	0.6931	0.6971	0.4915
	...	...	...	...
	14	0.1387	0.1184	0.9551
	15	0.1187	0.1186	0.9530
VGG-19	1	1.2353	215.4755	0.5085
	2	0.1476	5.3820	0.5662
	...	...	...	...
	14	0.0183	0.1078	0.9615
	15	0.0245	0.0629	0.9829
GoogleNet	1	0.2224	0.1397	0.9444
	2	0.2077	0.1697	0.9402
	...	...	...	...
	14	0.1155	0.1363	0.9423
	15	0.1047	0.1129	0.9637
Our model	1	0.9115	1.1247	0.5342
	2	0.4438	0.4807	0.8825
	...	...	...	...
	14	0.1665	0.3474	0.9551
	15	0.1558	0.3746	0.9880

**Table 3** Values of accuracy, precision, recall and F1-score

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
AlexNet	95.08	95.30	95.04	95.07
VGG-19	94.23	94.38	94.32	94.23
GoogleNet	95.51	97.86	95.54	95.49
Our Model	97.86	97.87	97.85	97.86

**Table 4** Comparative analysis of our work with other works using the dataset

S. No.	Work	Methodology used	Accuracy (%)	F1-score (%)
1	Lee et al. [21]	ResNet101V2	97	97
2	Varalakshmi et al. [22]	Inception V3	96	–
3	Gallego et al. [23]	U-Net	–	97.5
4	Proposed work	CNN model	97.86	97.86

In Table 2, it is observed that our model has outperformed the other three models and the validation accuracy achieved by the model is 98.80%.

The metrics used to evaluate individual models can be seen under in Eqs. (1–4), representing the accuracy, precision, recall, and F1-score.

$$\text{Accuracy} = \frac{p + r}{p + q + r + s} \quad (1)$$

$$\text{Precision} = \frac{p}{p + s} \quad (2)$$

$$\text{Recall} = \frac{p}{p + q} \quad (3)$$

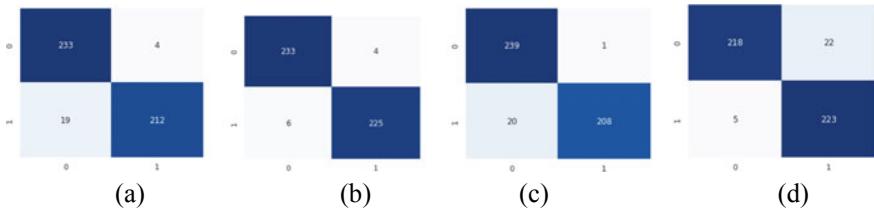
$$\text{F1 - Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

where p = true positive, q = false positive, r = true negative, and s = false negative.

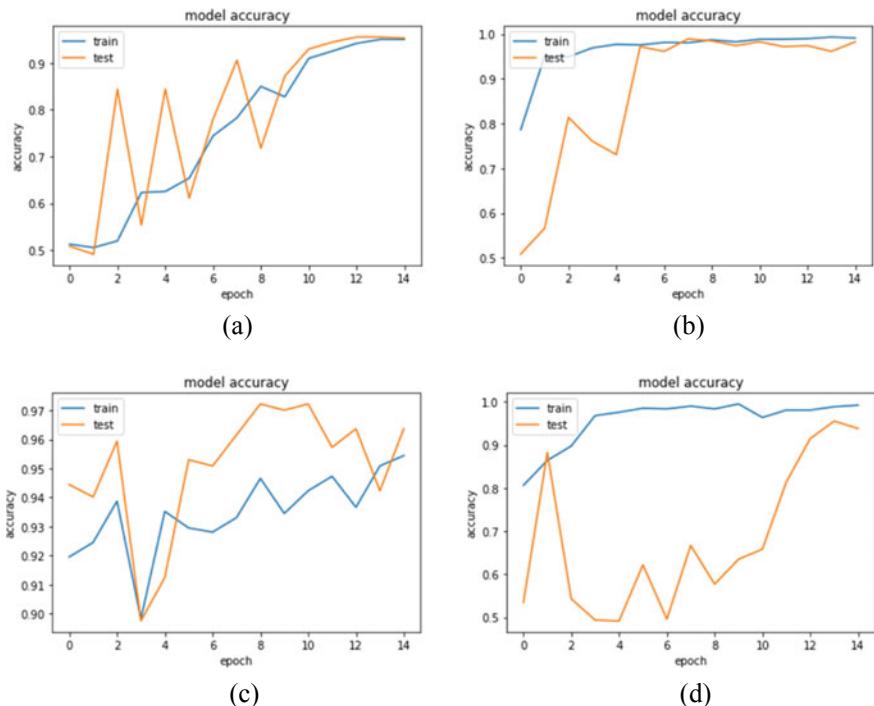
In Table 3, we can see the values of accuracy, precision, and recall of individual models in which AlexNet, VGG-19, GoogleNet, and our model achieved accuracies of the value of 95.08, 94.23, 95.51, and 97.86, respectively. Table 4 depicts the accuracy comparison of proposed model with some other models.

The confusion matrices of the models can be visualized in Fig. 6, the graph depicting the variation in the value of accuracy with epochs can be visualized in Fig. 7, and loss with epochs can be seen in Fig. 8.

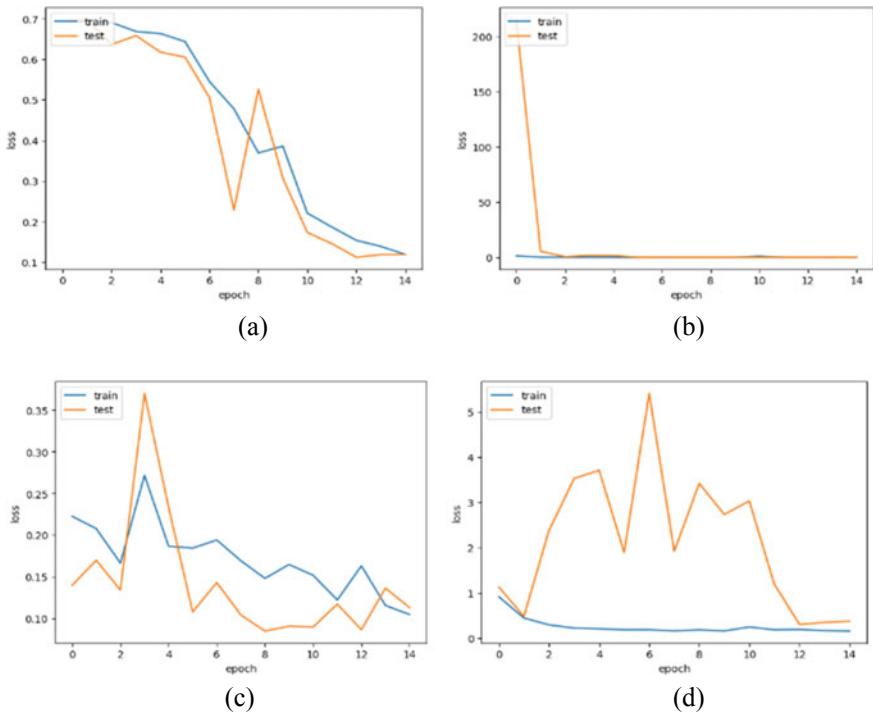
The confusion metrics that are represented in Fig. 6 clearly show the performance of the models on an individual level, the values of false positive (q) and false negative (s) is least for our model when compared with other models.



**Fig. 6** Confusion matrices of models: **a** AlexNet, **b** VGG-19, **c** GoogleNet, **d** Our model



**Fig. 7** Graph of accuracy with respect to epochs of models: **a** AlexNet, **b** VGG-19, **c** GoogleNet, **d** Our model



**Fig. 8** Graph of loss with respect to change in epochs of models: **a** Alexnet, **b** VGG-19, **c** GoogleNet, **d** Our model

## 6 Conclusion

The kidney is one of the main functional organs of the human body. Its care is very important for the well-being of an individual. The kidney glomerulus is one of the parts of the human kidney. It is responsible for the removal of the waste from the body, which makes its proper diagnosis a very important need. In this study, various CNN models for the classification of sclerosed and normal kidney glomerulus is used. The CNN used in this work are Alexnet, VGG-19, GoogleNet, and the customized CNN model and the various accuracies achieved by the models are 95.08, 94.23, 95.51, and 97.86, respectively. The study not only performs the classification of kidney glomeruli but also paves a way for further enhancements in the binary classification of kidney glomeruli. This work can further be enhanced by increasing the size of the data for better training of the models. The work has shown quite promising results and can be used in further research work also. To perform further investigations on this work, we can use various other models such as VGG-16 or ResNet-152. The models can also be evaluated further by using more optimizers for the classification. The performance can also be enhanced by increasing the size of the dataset either

by resampling or data augmentation. The proposed study may not always be fully accurate and may sometimes misclassify the kidney glomeruli.

## References

1. Nyengaard JR, Bendtsen TF (1992) Glomerular number and size in relation to age, kidney weight, and body surface in normal man
2. Bueno G, Gonzalez-Lopez L, Garcia-Rojo M, Laurinavicius A, Deniz O (2020) Data for glomeruli characterization in histopathological images. *Data Br* 29:105314. <https://doi.org/10.1016/j.dib.2020.105314>
3. Kaushik R, Kumar S, Pooling M (2019) Image segmentation using convolutional neural network. *Int J Sci Technol Res* 8(11). [Online]. Available: [www.ijstr.org](http://www.ijstr.org)
4. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 234–241
5. Rizwan I, Haque I, Neubert J (2020) Deep learning approaches to biomedical image segmentation. *Inf Med Unlocked* 18. <https://doi.org/10.1016/j.imu.2020.100297>
6. Jayapandian CP et al (2021) Development and evaluation of deep learning-based segmentation of histologic structures in the kidney cortex with multiple histologic stains. *Kidney Int* 99(1):86–101. <https://doi.org/10.1016/j.kint.2020.07.044>
7. Wang S, Yang DM, Rong R, Zhan X, Xiao G (2019) Pathology image analysis using segmentation deep learning algorithms. *Am J Pathol* 189(9):1686–1698. <https://doi.org/10.1016/j.ajpath.2019.05.007>
8. Bueno G, Fernandez-Carrobles MM, Gonzalez-Lopez L, Deniz O (2020) Glomerulosclerosis identification in whole slide images using semantic segmentation. *Comput Methods Programs Biomed* 184:105273. <https://doi.org/10.1016/j.cmpb.2019.105273>
9. Fu X, Liu T, Xiong Z, Smaill BH, Stiles MK, Zhao J (2018) Segmentation of histological images and fibrosis identification with a convolutional neural network. *Comput Biol Med* 98:147–158. <https://doi.org/10.1016/j.compbioemed.2018.05.015>
10. Kannan S et al (2019) Segmentation of glomeruli within trichrome images using deep learning. *Kidney Int Rep* 4(7):955–962. <https://doi.org/10.1016/j.ekir.2019.04.008>
11. Sun Y, Huang X, Zhou H, Zhang Q (2021) SRPN: similarity-based region proposal networks for nuclei and cells detection in histology images. *Med Image Anal* 72. <https://doi.org/10.1016/j.media.2021.102142>
12. Liu X, Guo Z, Cao J, Tang J (2021) MDC-net: a new convolutional neural network for nucleus segmentation in histopathology images with distance maps and contour information. *Comput Biol Med* 135. <https://doi.org/10.1016/j.compbioemed.2021.104543>
13. Chen X, Duan Q, Wu R, Yang Z (2021) Segmentation of lung computed tomography images based on SegNet in the diagnosis of lung cancer. *J Radiat Res Appl Sci* 14(1):396–403. <https://doi.org/10.1080/16878507.2021.1981753>
14. da Cruz LB et al (2020) Kidney segmentation from computed tomography images using deep neural network. *Comput Biol Med* 123. <https://doi.org/10.1016/j.compbioemed.2020.103906>
15. Heller N et al (2019) The KiTS19 challenge data: 300 kidney tumor cases with clinical context, CT semantic segmentations, and surgical outcomes, pp 1–14 [Online]. Available: <http://arxiv.org/abs/1904.00445>
16. Gong Z, Kan L (2021) Segmentation and classification of renal tumors based on convolutional neural network. *J Radiat Res Appl Sci* 14(1):412–422. <https://doi.org/10.1080/16878507.2021.1984150>
17. Manjunath RV, Kwadiki K (2022) Automatic liver and tumour segmentation from CT images using deep learning algorithm. *Results Control Optim* 6. <https://doi.org/10.1016/j.rico.2021.100087>

18. Zhao W, Jiang D, Peña Queralta J, Westerlund T (2020) MSS U-Net: 3D segmentation of kidneys and tumors from CT images with a multi-scale supervised U-Net. *Inform Med Unlocked* 19. <https://doi.org/10.1016/j.imu.2020.100357>
19. Anand V, Gupta S, Koundal D, Nayak SR, Barsocchi P, Bhoi AK (2022) Modified U-NET architecture for segmentation of skin lesion. *Sensors* 22(3). <https://doi.org/10.3390/s22030867>
20. de Bono B, Grenon P, Baldock R, Hunter P (2013) Functional tissue units and their primary tissue motifs in multi-scale physiology. *J Biomed Seman* 4(1):1–13. <https://doi.org/10.1186/2041-1480-4-22>
21. Lee H-C, Aqil AF (2022) Combination of transfer learning methods for kidney glomeruli image classification. *Appl Sci* 12:1040. <https://doi.org/10.3390/app12031040>
22. Varalakshmi P, Saroja S, Ketharaman S, Shimola S (2022) Glomeruli identification in renal biopsy using deep learning approaches. In: 2022 International conference on innovative computing, intelligent communication and smart electrical systems (ICSES), Chennai, India, pp 1–8. <https://doi.org/10.1109/ICSES55317.2022.9914279>
23. Gallego J, Swiderska-Chadaj Z, Markiewicz T, Yamashita M, Gabaldon MA, Gertych A (2021) A U-Net based framework to quantify glomerulosclerosis in digitized PAS and H&E stained human tissues. *Comput Med Imaging Graph* 89:101865. Epub: 2021 Jan 28. PMID: 33548823. <https://doi.org/10.1016/j.compmedimag.2021.101865>

# Application of Modified Differential Evolution Technique for Automatic Generation Control Problem



Asish Kumar Panigrahi, Rabindra Kumar Sahu,  
and Tulasichandra Sekhar Gorripotu

**Abstract** In this article, the outcome of unified power flow controller (UPFC) is observed. Initially, a two area six unit power system is considered for examination. Thermal, hydro and wind units are assumed in area-1 and thermal, hydro and diesel units are in area-2. The developed model is composed of a hybrid control scheme of fractional order proportional integral derivative controller (FOPID) and modified differential evolution (MDE) algorithm. The effectiveness of the proposed technique is tested for variations in the step load. From the analysis, it is confirmed that FOPID plus UPFC employed system working effectively.

**Keywords** FOPID · MDE · UPFC

## 1 Introduction

In present days, it was well-known that automatic load frequency control (ALFC) is an important challenge because the power system network is spread out over a large area and different areas are linked together. ALFC is a power sharing mechanism that maintains a consistent operating frequency across the system to regulate the power among neighboring generating areas. Without a tie-line link, this sharing would be impossible. The system frequency adjusts in response to an abrupt change in the connected load. However, it is a requirement to keep the system running at a constant frequency during the operation, which is accomplished using tie-lines to share power among the interconnected systems [1–3]. It's challenging to keep the frequency at a consistent level while also keeping the power allocation in the tie-lines

---

A. K. Panigrahi · R. K. Sahu

Department of Electrical and Electronics Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla, Odisha 768018, India

T. S. Gorripotu (✉)

Department of Electrical and Electronics Engineering, Sri Sivani College of Engineering, Srikakulam, Andhra Pradesh 532410, India

e-mail: [gtchsekhar@gmail.com](mailto:gtchsekhar@gmail.com); [gtchsekhar@srисivani.com](mailto:gtchsekhar@srисivani.com)

within allowed limits. The automatic generation control (AGC) concept can help to overcome the problems pointed above. The frequency of the electrical power output, with the exchange of scheduled power among the related locations, will be controlled by the mechanical power input to the generator.

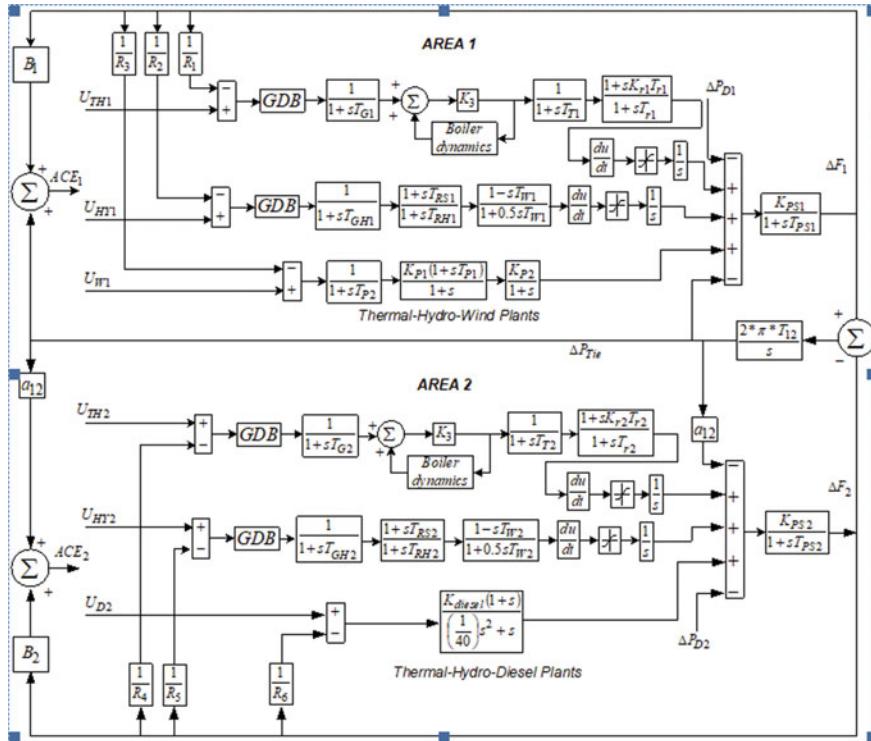
## 2 Literature Review and Proposed System

It was well-known that, in past, researchers worked on the concept of AGC and suggested different types of controllers and techniques [4–6]. However, most of the researchers confined to classical controllers and its derivates. Naga Sai Kalyan et al. [7] have implemented PI, PD, PID and PIDN controllers for the multiarea power system. Nayak et al. [8] have suggested conventional controllers for 3-area 6-unit system. They have employed genetic algorithm (GA), particle swarm Optimization (PSO) and grasshopper optimization algorithm (GOA) for getting optimal values. The authors also suggested a new controller called PDF plus (1 + PI) for AGC concept which provides better results. Khadum et al. [9] have investigated 2-area power system for frequency control by employing PSO technique-based PID controller. From this discussion, it has been examined that, there is a possibility of proposing new controller and techniques for the concept of AGC.

In this work, 2-area 6-unit dissimilar source units are considered. Thermal, hydro and wind plants in area-1 and thermal, hydro and wind units are kept in area-2 as shown in Fig. 1. Both the areas are equal in rating and capacity. Here, dynamics of boiler are considered and designed in MATLAB software. GDB and GRC are also included to have realistic study. The system is designed under SIMULINK background and appropriate participation factors are considered for proper utilization of renewable energy sources.

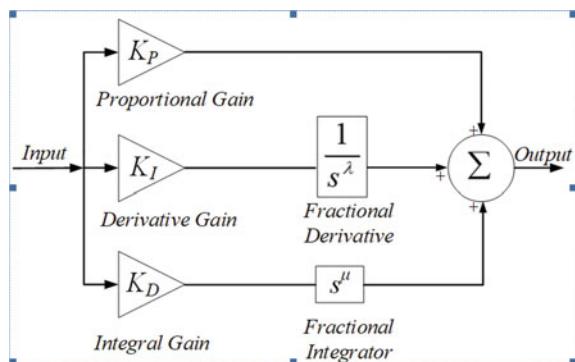
## 3 Controller Structure

The secondary controller is essential for the regulation of power system parameters. There are several derivative controllers of classical PID and lot of research has been done on it. The presentation of classical controllers is not much effective for nonlinear models and higher order systems. The above disadvantages can be overcome by having fractional order PID (FOPID) controller. It is evolved by placing an integrator and differentiator at the appropriate positions for PID controller and it is shown in Fig. 2. The main advantages of FOPID are wide operating range and less sensitivity. When compared with PID, it is having extra two parameters to tune so that, FOPID provides better results [10, 11]. Keeping all in view, FOPID controller is selected to solve AGC problem.



**Fig. 1** Block diagram of the proposed system

**Fig. 2** FOPID structure



## 4 Analytical Analysis

Initially, a two-area power system having six units is considered for analysis. In area-1, thermal, hydro and wind systems are incorporated. In area-2, thermal, hydro and diesel are considered. Appropriate area participation factors (apfs) are established

for unit. The apf of units are: thermal = 0.575, hydro = 0.3 and wind = diesel = 0.125. The system is tested for 1% step load perturbation in area-1 by having FOPID as controller. In the next step, for further improvement of dynamic responses, unified power flow controller (UPFC) is incorporated in area-1. The UPFC is a FACT device which injects active and reactive powers simultaneously into the system [6]. The control parameters are tuned by using modified differential evolution technique. The process of tuning is clearly explained in [12–14].

### With FOPID:

$$\begin{aligned} K_{II} &= 0.8504; \quad K_{I2} = 0.4945; \quad K_{I3} = 0.9482; \quad K_{I4} = 0.7978; \quad K_{I5} = 0.1330; \quad K_{I6} = 0.2378; \\ K_{D1} &= 0.1277; \quad K_{D2} = 0.5341; \quad K_{D3} = 0.0567; \quad K_{D4} = 0.4686; \quad K_{D5} = 0.9827; \quad K_{D6} = 0.6822; \\ K_{P1} &= 0.8709; \quad K_{P2} = 0.4962; \quad K_{P3} = 0.2877; \quad K_{P4} = 0.1830; \quad K_{P5} = 0.9612; \quad K_{P6} = 0.7563; \\ N_{C1} &= 0.9504; \quad N_{C2} = 0.6713; \quad N_{C3} = 0.1861; \quad N_{C4} = 0.2898; \quad N_{C5} = 0.0582; \quad N_{C6} = 0.0441; \\ N_{C7} &= 0.9504; \quad N_{C8} = 0.6713; \quad N_{C9} = 0.1861; \quad N_{C10} = 0.2898; \quad N_{C11} = 0.0582; \quad N_{C12} = 0.0441; \\ ITAE &= 1.1342; \end{aligned}$$

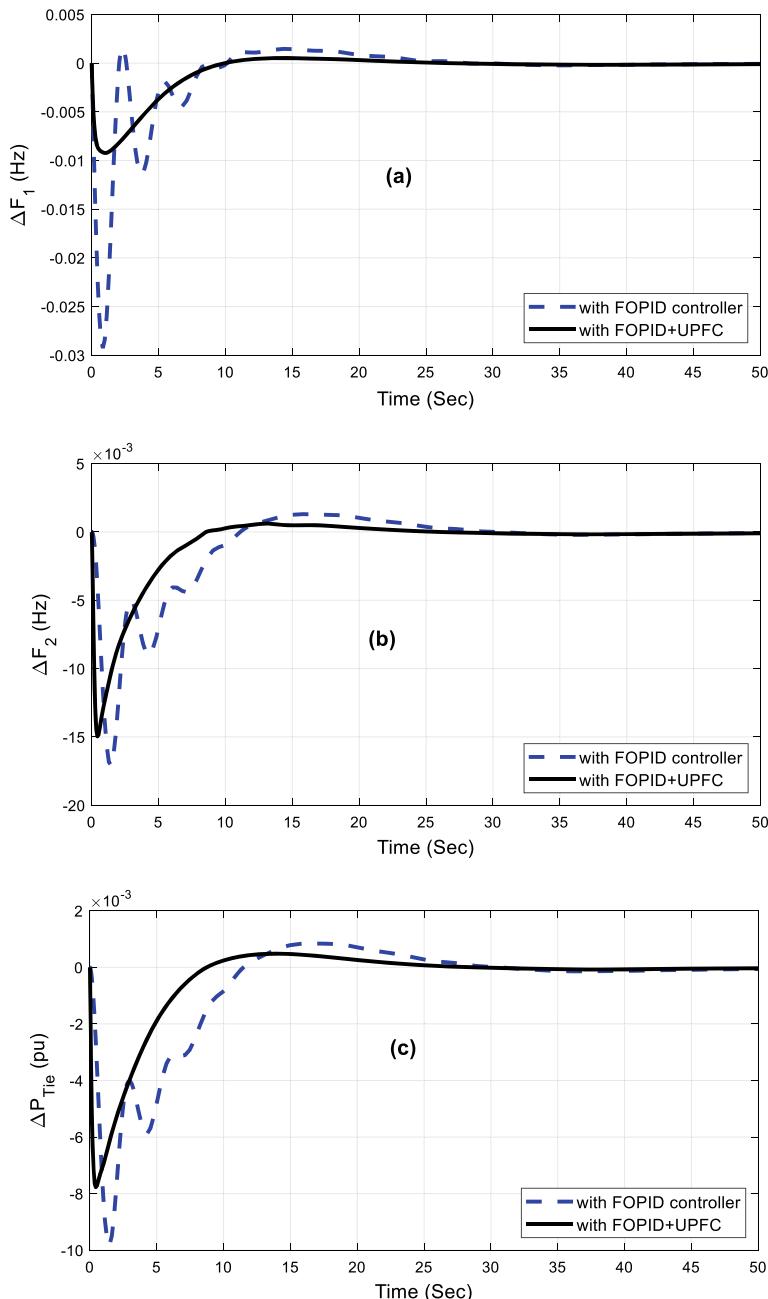
### With FOPID plus UPFC:

$$\begin{aligned} K_{II} &= 0.6295; \quad K_{I2} = 0.0607; \quad K_{I3} = 0.6740; \quad K_{I4} = 0.4774; \quad K_{I5} = 0.3055; \quad K_{I6} = 0.5163; \\ K_{D1} &= 0.7070; \quad K_{D2} = 0.8136; \quad K_{D3} = 0.3158; \quad K_{D4} = 0.3113; \quad K_{D5} = 0.3450; \quad K_{D6} = 0.6663; \\ K_{P1} &= 0.8611; \quad K_{P2} = 0.7618; \quad K_{P3} = 0.8758; \quad K_{P4} = 0.8712; \quad K_{P5} = 0.1728; \quad K_{P6} = 0.8502; \\ N_{C1} &= 0.9596; \quad N_{C2} = 0.7702; \quad N_{C3} = 0.8750; \quad N_{C4} = 0.0674; \quad N_{C5} = 0.6468; \quad N_{C6} = 0.3241; \\ N_{C7} &= 0.9596; \quad N_{C8} = 0.7702; \quad N_{C9} = 0.8750; \quad N_{C10} = 0.0674; \quad N_{C11} = 0.6468; \quad N_{C12} = 0.3241; \\ ITAE &= 0.7430 \end{aligned}$$

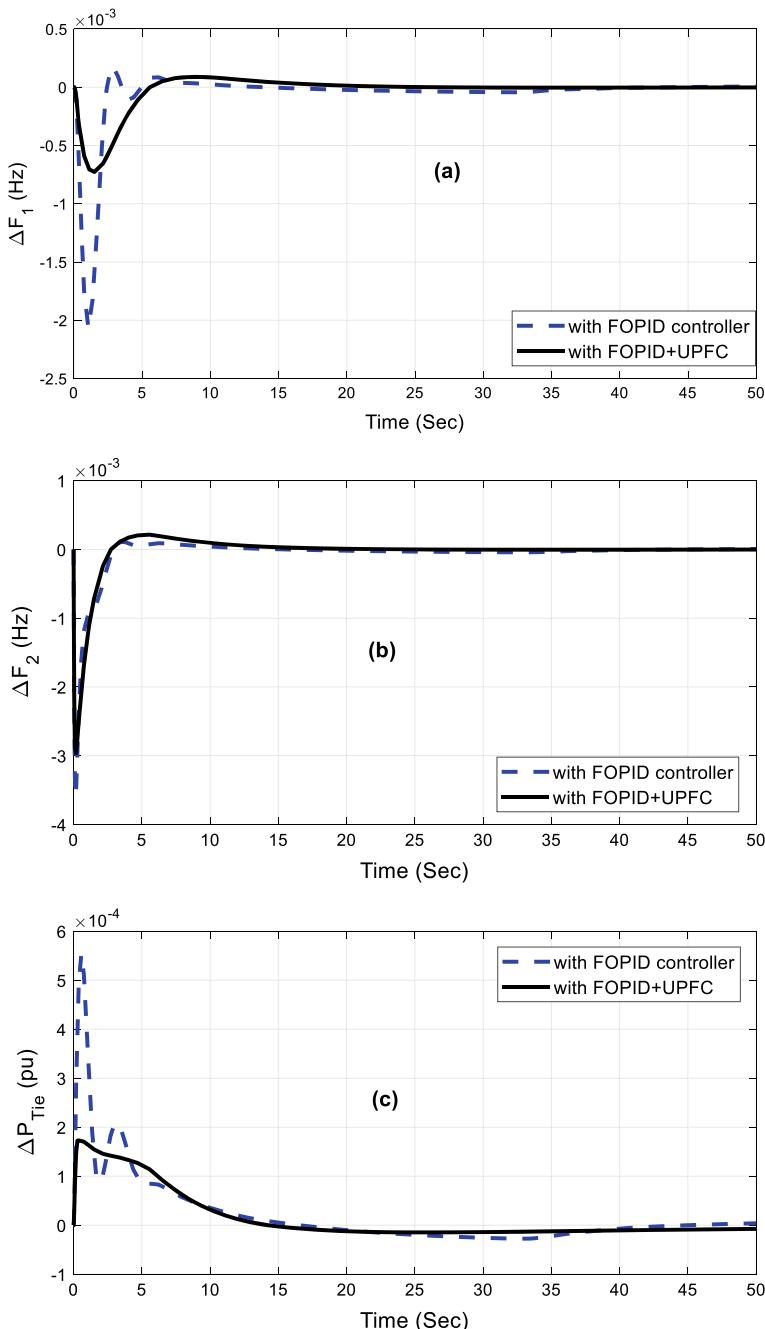
Assume that, there will be 1% disturbance in area-1 which starts at  $t = 0$  s. Due to this, the system dynamics will change and FOPID controller is kept in closed loop system to minimize the errors. Better transient responses are observed with FOPID and those responses are further improved by integrating UPFC. The related responses can be observed in Fig. 3a–c. In the next step, the robustness is verified by interchanging disturbance to area-2. Though the controller parameters are unchanged, the proposed FOPID provides good results. This can be observed from the Fig. 4a–c. Lastly, disturbance is applied in both areas and the responses are plotted as shown in Fig. 5a–c.

## 5 Conclusion

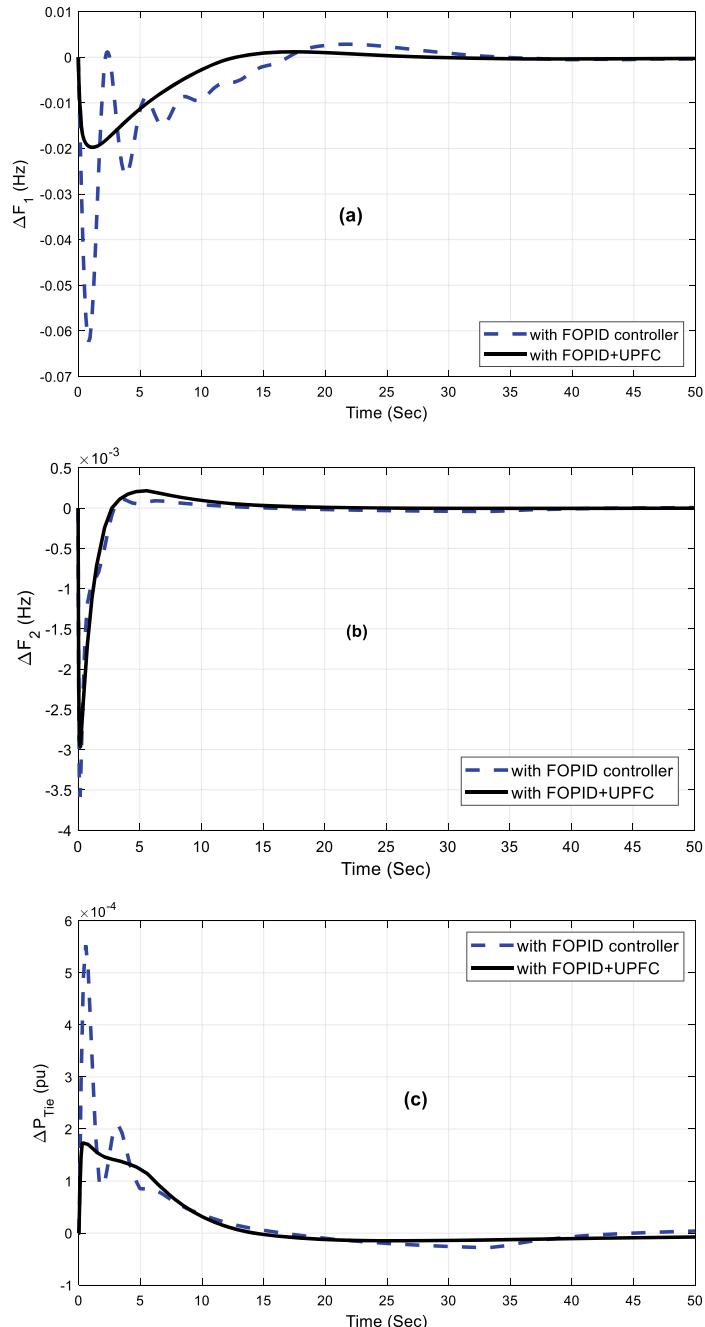
In this present study, the advanced approach of MDE technique for AGC concept is implemented. A 2-area and 6-unit power plants are considered and FOPID controller is incorporated in the system. The UPFC is also included in the tie-line to improve the system performance. The results clarified that, FOPID + UPFC system achieved better results in all aspects such as settling time, over and under shoots. The FOPID controller works well for nonlinear system and provides better transient responses compared with other classical controllers.



**Fig. 3 a–c:** Responses for 1% SLP area-1



**Fig. 4 a–c:** System responses for 1% SLP in area-2



**Fig. 5 a–c:** System responses for 1% SLP in both areas

## References

1. Elgerd OI (2000) Electric energy systems theory—an introduction. Tata McGraw Hill, New Delhi
2. Bevrani H (2014) Robust power system frequency control, vol 4. Springer, New York. <https://doi.org/10.1007/978-3-319-07278-4>
3. Bervani H, Hiyama T (2011) Intelligent automatic generation control. CRC Press
4. Chandra Sekhar GT, Vijaya Kumar D, Manamadha Kumar B, Ramana P (2018) Design and analysis of BFOA optimized PID controller with derivative filter for frequency regulation in distributed generation system. *Int J Autom Control* 12(2):291–323
5. Peddakapu K, Mohamed MR, Srinivasarao P, Arya Y, Leung PK, Kishore DJK (2022) A state-of-the-art review on modern and future developments of AGC/LFC of conventional and renewable energy-based power systems. *Renew Energy Focus*
6. Sahu RK, Chandra Sekhar GT, Panda S (2015) A hybrid DE-PS algorithm for load frequency control under deregulated power system with UPFC and RFB. *Ain Shams Eng J* 6:893–911
7. Naga Sai Kalyan CH, Suresh CV (2022) Performance evaluation of various traditional controllers in automatic generation control of multi-area system with multi-type generation units. In: Smart and intelligent systems: proceedings of SIS 2021. Springer Singapore, pp 395–404
8. Nayak PC, Prusty RC, Panda S (2021) Grasshopper optimization algorithm optimized multi-stage controller for automatic generation control of a power system with FACTS devices. *Protect Control Modern Power Syst* 6:1–15
9. Kadhum AA, Sahib TM, Ali MMM (2019) Particle swarm optimization algorithm based PID controller for the control of the automatic generation control. Recent trends and advances in artificial intelligence and internet of things. Springer International Publishing, Cham, pp 217–225
10. Arya Y (2020) A novel CFFOPI-FOPID controller for AGC performance enhancement of single and multi-area electric power systems. *ISA Trans* 100:126–135
11. Kumar N, Tyagi B, Kumar V (2018) Application of fractional order PID controller for AGC under deregulated environment. *Int J Autom Comput* 15:84–93
12. Storn R (1995) Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical report. International Computer Science Institute, p 11
13. Wang Z, Chen Z, Wang Z, Wei J, Chen X, Li Q, Sheng W (2022) Adaptive memetic differential evolution with multi-niche sampling and neighborhood crossover strategies for global optimization. *Inf Sci* 583:121–136
14. Chintu JMR, Sahu RK, Panda S (2022) Adaptive differential evolution tuned hybrid fuzzy PD-PI controller for automatic generation control of power systems. *Int J Ambient Energy* 43(1):515–530

# A Four-Valued Epistemic Logic for Metadata Modelling from Medical Articles on Pain Therapies



Simone Cuconato

**Abstract** In this paper, a four-valued epistemic logic for metadata modelling from medical articles on pain therapies has been proposed. The advantage of using a four-valued epistemic logic is that it allows us to reason with partial, incomplete or even incoherent information. This is especially the case in the information sciences, which have to deal with an extremely varied quantity and quality of data. The application of logic to metadata extraction provides a powerful and rigorous tool capable of assessing the quality of the extracted information; that is, establishing the correctness of the extracted information.

**Keywords** Epistemic logic · Many-valued logics · Metadata modelling

## 1 Introduction

The theoretical foundation of this paper is the belief that “data science performs at its best when coupled with the subtle art of modelling” [7]. Over the past two decades, “the role of models and modelling in scientific research has been studied exhaustively by philosophers of science” [12]. It is safe to affirm that a consensus exists not only that models can be used to explain the physical world [16], but also to understand the world of data [7]. Assessing data quality is a challenging task. Specifically, assessing the quality of extracted metadata means determining the fitness for purpose of a given metadata extraction system.

This paper has two goals. The first goal is to use four-valued epistemic logic (FVEL, for short) to model automatically extracted structured metadata. The second goal is to investigate how to model the extracted metadata through a specific application case: metadata extracted from scientific articles on pain therapies.

---

S. Cuconato (✉)

Department of Informatics, Modeling, Electronics and Systems Engineering, University of Calabria, Cosenza, Italy

e-mail: [simone.cuconato@unical.it](mailto:simone.cuconato@unical.it)

Institute of Informatics and Telematics (IIT)-CNR, Pisa-Cosenza, Italy

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

631

A. K. Das et al. (eds.), *Computational Intelligence in Pattern Recognition*,

Lecture Notes in Networks and Systems 725,

[https://doi.org/10.1007/978-981-99-3734-9\\_51](https://doi.org/10.1007/978-981-99-3734-9_51)

Metadata [8] play a fundamental role in data science, as they provide a *criterion of identity*<sup>1</sup> for data. Inspired in Willard Van Orman Quine's well-known slogan "No entity without identity" [10], I could say "No data without metadata". Over the last twenty years, several systems have been developed to automatically extract metadata from scientific articles. In this paper, through epistemic logic, a modelling of the metadata extracted by one of the best systems for the automatic extraction of metadata from medical/scientific articles: CERMINE<sup>2</sup> has been proposed. The main objective of my contribution is to provide a logic model that can evaluate the quality and correctness of the extracted information and support the research community in building solutions to pain therapies.

## 2 A Four-Valued Epistemic Logic for Metadata Modelling

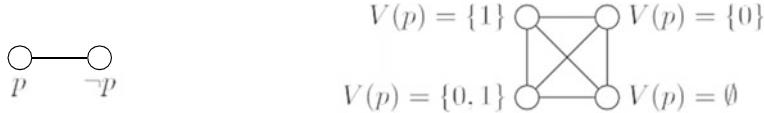
Epistemic logic is an extension of classical logic that has as its object of study the statements of belief and knowledge. Standard epistemic logic [15] is normally used "to model two aspects of a situation: the factual and the epistemic aspects" [11]. The truth, though, is not always approachable and in many cases, it is necessary to reason with partial, incomplete or even incoherent information. This is especially the case in the information sciences, which have to deal with an extremely varied quantity and quality of data. Assuming a minimal definition of information as "data + semantics", trust on extracted information can be identified with the result of a consistency assessment. In this context, an extracted information is consistent when it allows preserving: (i) the set of beliefs and knowledge base of the extraction agent; and (ii) the informational properties of the object from which the extraction was performed.

Therefore, having to deal with sets of beliefs and knowledge, epistemic logic turns out to be the most suitable logic for this task. In this section, I describe a four-valued epistemic logic designed to deal with these situations.<sup>3</sup> The advantage of using a four-valued epistemic logic is that it does not leave "out of the discussion an important factor in the formation of beliefs: *evidence*" [11]. Nuel Belnap [1] first, and later Jon Michael Dunn [4] and Graham Priest [9], provided an initial interpretation of a four-valued logic, centred precisely on the notion of evidence. In that logic, a proposition  $p$  can be, besides *true* or *false*, *both* (true and false) or *neither* (true nor false).

<sup>1</sup> A criterion of identity is generally understood as a principle that is supposed to account for the identity of a thing.

<sup>2</sup> CERMINE "is a comprehensive open-source system for extracting structured metadata from scientific articles in a born-digital form" [13, 14] (in particular, PDF documents).

<sup>3</sup> For an application of standard epistemic logic to metadata modelling see [2, 3]. Compared to standard epistemic logic, four-valued epistemic logic allows more precise analysis extracted information. A four-valued epistemic logic is a more precise tool for understanding the extracted metadata because allows us to reason with partial, incomplete or incoherent information. This will become evident in Sect. 3 when I consider a *partially correct* metadata.



**Fig. 1** A standard epistemic model (left) and a non-standard FVEL (right)<sup>7</sup>

Specifically, I apply a simplified version of the four-valued epistemic logic developed by Yuri David Santos [11], to the modelling of metadata. At the syntactic level, a particular kind of proposition  $p_{\mathcal{E}}$  is used as shown in Eq. (1):

$$p_{\mathcal{E}} =_{\text{def}} \mathcal{E}_{m_i}^{d_i} \quad (1)$$

where  $\mathcal{E}_{m_i}^{d_i}$  read as “extracts metadata  $m_i$  from document  $d_i$ ”.

Also, it is important to specify that: (i) this language includes the epistemic constructions  $K_a\varphi$  (read as “agent  $a$  knows (that)  $\varphi$ ”); and (ii) an agent should be understood as a “Metadata Extraction Agent” (MEA).<sup>4</sup>

**Definition 1** [Syntax of  $\mathcal{L}_{\text{FV}}$ ]<sup>5</sup> Let  $\mathcal{P}$  be a countable set of atomic propositions and  $\mathcal{A}$  a finite set of agents, a well-formed formula  $\varphi$  in language  $\mathcal{L}_{\text{FV}}$  is inductively defined as shown in Eq. (2):

$$\varphi := p_{\mathcal{E}} | \neg\varphi | \sim\varphi | \varphi \wedge \varphi | K_a\varphi \quad (2)$$

with  $p_{\mathcal{E}} \in \mathcal{P}$  and  $a \in \mathcal{A}$ .

In accordance with [2, 3], at the “semantic level the concept of possible world will be replaced with that of *possible extraction*” [2].

**Definition 2** [Four-Valued Epistemic Model]<sup>6</sup> “Given a set  $\mathcal{P}_{\mathcal{E}}$  of primitive propositions and a non-empty finite set  $\mathcal{F}$  of MEA” [2], an interpretation  $M_{\text{FV}} : \langle E, R_{\text{FV}}^{\mathcal{F}}, V_{\text{FV}}^{\mathcal{P}_{\mathcal{E}}} \rangle$  where:

- $E \neq \emptyset$  “is a set of possible extractions” [2];
- $R_{\text{FV}}^{\mathcal{F}} = (R_{\text{FV}1}^{\mathcal{F}}, R_{\text{FV}2}^{\mathcal{F}}, \dots, R_{\text{FV}n}^{\mathcal{F}})$  is an  $n$ -tuple of binary relations on  $E$ ;
- $V_{\text{FV}}^{\mathcal{P}_{\mathcal{E}}} : \mathcal{P}_{\mathcal{E}} \times E \rightarrow 2^{\{0,1\}}$ : “is a valuation function that, assigns to each proposition one of four truth values: {0} is *false* (*f*), {1} is *true* (*t*),  $\emptyset$  is *none* (*n*) and {0, 1} is *both* (*b*)” [11].

Figure 1 compares a standard epistemic model with a non-standard four-valued epistemic model, where {1}, {0}, {0, 1}, and  $\emptyset$  mean, in order, *true*, *false*, *both*, and *none*.

<sup>4</sup> See [2].

<sup>5</sup> See [11].

<sup>6</sup> See [2].

<sup>7</sup> Figure taken from [11].

**Definition 3** [Semantics of Syntax of  $\mathcal{L}_{FV}$ ] With  $p_{\mathcal{E}} \in \mathcal{P}$ ,  $e \in E$ ,  $a \in \mathcal{A}$ , and  $\varphi, \psi \in \mathcal{L}_{FV}$ , the satisfaction relation  $\models$  is inductively defined as shown in Eqs. (3)–(10):

$$M_{FV}, e \models p_{\mathcal{E}} \text{ iff } 1 \in V_{FV}^{\mathcal{P}_{\mathcal{E}}}(p_{\mathcal{E}}, e) \quad (3)$$

$$M_{FV}, e \models \neg p_{\mathcal{E}} \text{ iff } 0 \in V_{FV}^{\mathcal{P}_{\mathcal{E}}}(p_{\mathcal{E}}, e) \quad (4)$$

$$M_{FV}, e \models \varphi \wedge \psi \text{ iff } M_{FV}, e \models \varphi \text{ and } M_{FV}, e \models \psi \quad (5)$$

$$M_{FV}, e \models \neg(\varphi \wedge \psi) \text{ iff } M_{FV}, e \models \neg\varphi \text{ or } M_{FV}, e \models \neg\psi \quad (6)$$

$$M_{FV}, e \models \sim \varphi \text{ iff } M_{FV}, e \not\models \varphi \quad (7)$$

$$M_{FV}, e \models \neg \sim \varphi \text{ iff } M_{FV}, e \models \varphi \quad (8)$$

$$M_{FV}, e \models \neg \neg \varphi \text{ iff } M_{FV}, e \models \varphi \quad (9)$$

$$M_{FV}, e_1 \models K_a \varphi \text{ iff for all } e_2 \in E \text{ such that } e_1 R e_2, \text{ it holds that } M_{FV}, e_2 \models \varphi \quad (10)$$

Since the interpretation of formulas is based on the notion of evidence, it is necessary to make some clarifications at the semantic level. Firstly, non-epistemic “formulas  $\varphi$  and  $\neg\varphi$  are read as *there is evidence for  $\varphi$*  and *there is evidence against  $\varphi$* , respectively” [11]. Secondly, “the negation  $\sim$  is classical:  $\sim \varphi$  means that *it is not the case that  $\varphi$* ” [11]. Thirdly, the  $K$  operator cannot be read in the standard way as “*in all possible worlds compatible with what  $a$  knows, it is the case that  $\varphi$* ” [6], but rather as:

$K_a \varphi$ : agent  $a$  knows that there is evidence for  $\varphi$ .

**Definition 4** [Extended Evaluation Function] The *extended evaluation function*  $\overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}} : \mathcal{L}_{FV} \times E \rightarrow 2^{\{0,1\}}$  is depicted as follows:

$$1 \in \overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\varphi, e) \text{ iff } M_{FV}, e \models \varphi$$

$$0 \in \overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\varphi, e) \text{ iff } M_{FV}, e \models \neg\varphi$$

As Santos specifies “since the semantics of FVEL is non-compositional, the readings of its formulas will be non-compositional as well. Truth and falsity of formulas are evaluated independently, and for that reason the semantic conditions for each negated formula are defined separately” [11]. However, “even if the semantics of  $\neg$  is defined on a case-by-case, the connective is still truth-functional” [11].

**Table 1** For  $\neg\varphi$ <sup>8</sup>

$\neg\varphi$	
<i>t</i>	<i>f</i>
<i>f</i>	<i>t</i>
<i>n</i>	<i>n</i>
<i>b</i>	<i>b</i>

**Table 2** For  $\sim\varphi$ 

$\sim\varphi$	
<i>t</i>	<i>f</i>
<i>f</i>	<i>t</i>
<i>n</i>	<i>t</i>
<i>b</i>	<i>f</i>

**Table 3** For  $\varphi \wedge \psi$ 

$\varphi \wedge \psi$	<i>t</i>	<i>f</i>	<i>n</i>	<i>b</i>
<i>t</i>	<i>t</i>	<i>f</i>	<i>n</i>	<i>b</i>
<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>	<i>f</i>
<i>n</i>	<i>n</i>	<i>f</i>	<i>n</i>	<i>f</i>
<i>b</i>	<i>b</i>	<i>f</i>	<i>f</i>	<i>b</i>

Therefore, one must “think of the (four-valued) valuation function as representing evidence or information, while the accessibility relations account for the uncertainty of the agents about which evidential state is the correct one” [11].

**Definition 5** [Truth Table] Given any formula,  $\varphi$ , and any interpretation,  $V_{FV}^{\mathcal{P}_{\mathcal{E}}}$ , there are four possibilities:  $\varphi$  is true and not false,  $\varphi$  is false and not also true,  $\varphi$  is true and false,  $\varphi$  is neither true nor false. The truth conditions for the connectives  $\neg$ ,  $\sim$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$  are depicted as follows (Tables 1, 2, 3, 4 and 5):

**Definition 6** [Validity] A formula  $\varphi$  is valid if and only if  $M_{FV} \models \varphi$  for all models  $M_{FV}$ .

**Definition 7** [Four-Valued Epistemic Metadata Extraction Structure]<sup>9</sup> “A  $\mathcal{S}$  structure is of the form  $\mathcal{S} = \langle \mathcal{F}, E, \mathcal{P}_{\mathcal{E}}, M, D \rangle$ , where” [2]:

<sup>8</sup> Example for Table 1: “ $\overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\varphi, e) = \{0, 1\}$  iff  $0 \in \overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\varphi, e)$  and  $1 \in \overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\varphi, e)$  iff  $M_{FV}, e \models \neg\varphi$  and  $M_{FV}, e \models \varphi$  iff  $M_{FV}, e \models \neg\varphi$  and  $M_{FV}, e \models \neg\neg\varphi$  iff  $1 \in \overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\neg\varphi, e)$  and  $0 \in \overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\neg\varphi, e)$  iff  $\overline{V}_{FV}^{\mathcal{P}_{\mathcal{E}}}(\neg\varphi, e) = \{0, 1\}$ ” [11].

<sup>9</sup> See [2, 3].

**Table 4** For  $\varphi \vee \psi$ 

$\varphi \vee \psi$	$t$	$f$	$n$	$b$
$t$	$t$	$t$	$t$	$t$
$f$	$t$	$f$	$n$	$b$
$n$	$t$	$n$	$n$	$t$
$b$	$t$	$b$	$t$	$b$

**Table 5** For  $\varphi \rightarrow \psi$ 

$\varphi \rightarrow \psi$	$t$	$f$	$n$	$b$
$t$	$t$	$n$	$n$	$t$
$f$	$t$	$t$	$t$	$t$
$n$	$t$	$f$	$n$	$b$
$b$	$t$	$b$	$t$	$b$

$\mathcal{F} = \{a, b, c, \dots\}$  is a non-empty finite set of MEA,

$E = \{e_1, \dots, e_m\}$  is a non-empty set of possible extractions ( $|E| = m \in \mathbb{N}$ ),

$\mathcal{P}_E = \{p_{\mathcal{E}_1}, \dots, p_{\mathcal{E}_m}\}$  is a non-empty set of propositions ( $|\mathcal{P}_E| = m \in \mathbb{N}$ ),

$M = \{m_1, \dots, m_m\}$  is a non-empty set of metadata ( $|M| = m \in \mathbb{N}$ ),

$D = \{d_1, \dots, d_m\}$  is a non-empty set of documents ( $|D| = m \in \mathbb{N}$ ).

“ $\mathcal{S}$  is a structure in which possible extractions  $E$  occur.  $\mathcal{F}$  is the set of MEA, while  $\mathcal{P}_E$  is the set of epistemic propositions.  $M$  is the set of metadata and  $D$  is the set of documents” [2] (in this case, articles on pain therapy).

Since  $p_E$  denoted by  $\{1\}, \{0\}, \{0, 1\}$  and  $\emptyset$ , the fact that the information is, respectively, *true*, *false*, *both*, and *none*

$$\mathcal{E}_{m_i}^{d_i} = \{1\}/\{0\}/\{0, 1\}/\emptyset$$

### 3 An Application Case

In this section, how to model metadata extracted from scientific articles on pain therapies by applying FVEL has been presented. Extraction focuses on three specific descriptive metadata: title, author, and keywords. The document  $d_1$  [5] concerns the main approaches for pain management in hemophilic arthropathy. “Hemophilic arthropathy is a systemic arthropathy most commonly caused by hemophilia and characterized by repetitive hemarthroses and progressive joint disease” [5]. Consider the following structure  $\mathcal{S}_1 = \langle \mathcal{F}, E, \mathcal{P}_E, M, D \rangle$ :

$$\mathcal{F} = \{a\};$$

$$E = \{e_1\};$$

$$\mathcal{P}_{\mathcal{E}} = \{p_{\mathcal{E}_1}, p_{\mathcal{E}_2}, p_{\mathcal{E}_3}\};$$

$$M = \{m_1, m_2, m_3\};$$

$$D = \{d_1\}$$

Given the document  $d_1$ , MEA  $a$  (CERMINE) extracts ( $e_1$ ) the metadata  $m_1 =$  title,  $m_2 =$  author,  $m_3 =$  keywords. In Fig. 2, the metadata “title” is highlighted in red, the “author” metadata in yellow, and the metadata “keywords” in green.

In detail, in  $e_1$   $a$  extracts:

$$\mathcal{E}_{m_1}^{d_1} = \{1\}$$

$$\mathcal{E}_{m_2}^{d_1} = \{0, 1\}$$

$$\mathcal{E}_{m_3}^{d_1} = \{1\}$$

Figure 3 shows the extraction metadata results formatted in HTML form.

In particular, about the “Author” metadata ( $m_2$ ), the extracted information is partially correct because, on the one hand, it is true that the author’s name is reported correctly, but on the other hand, additional information is reported that is not part of the author’s name (such as *affiliation*). In this way, a FVEL-based model makes it possible to retain part of the extracted information without necessarily having to consider the extraction of the “Author” metadata completely wrong.

## 4 Conclusions and Future Work

There is no doubt that the potential of data science is increasingly being recognized. However, “one should be mindful that data without a model is just noise” [7]. For this reason, the development of data science must involve not only core disciplines such as computer science or statistics, but also logic, ethics, social sciences, and medical sciences. Motivated by the preceding concerns and observations, in this paper I moved within an interdisciplinary and multidisciplinary perspective in the fields of applied logic, data science and knowledge engineering, and I proposed a four-valued epistemic logic for modelling metadata from medical articles on pain therapies. In particular, the application of logic to metadata extraction provides a powerful and rigorous tool capable of assessing the quality of the extracted information, and assessing the quality of extracted metadata means determining the fitness for purpose



REVIEW

## Current and Emerging Approaches for Pain Management in Hemophilic Arthropathy

Roberta Gualtierotti · Francesco Tafuri · Sara Arcudi ·  
Pier Luigi Solimeno · Jacopo Acquati · Laura Landi · Flora Peyvandi

Received: August 7, 2021 / Accepted: December 7, 2021 / Published online: January 12, 2022  
© The Author(s) 2022

### ABSTRACT

**Introduction:** Hemophilia is an inherited bleeding hematological disorder characterized by the partial or complete deficiency of clotting factor VIII or IX. Hemophilic arthropathy is the consequence of repeated joint bleeding (hemarthrosis) and its management is based on the prevention of acute bleeding through the administration of the deficient clotting factor concentrate or non-factor therapies. In addition, the management of acute and chronic pain is pivotal in hemophilic arthropathy in order to restore function and allow rehabilitation of the joint.

**Methods:** We conducted a qualitative review of the literature regarding current and emerging strategies for pain treatment in hemophilic arthropathy. This review considers systemic and local pharmacological and non-pharmacological interventions for acute and chronic pain management.

**Results:** In hemophilic arthropathy, pain management is based on analgesics such as paracetamol, which represents the first choice for acute and chronic pain in adults and children, in association with opioids for adults. Non-steroidal anti-inflammatory drugs inhibit platelet function, so that the currently preferred drugs are short courses of cyclooxygenase 2 inhibitors. Local treatment with intra-articular injections of corticosteroids is an option for refractory cases and physiotherapy has an important role after hemarthrosis and for the long-term management of chronic pain for both pediatric and adult patients.

**Conclusions:** The management of pain in hemophilia requires more standardization. Meanwhile, the safest drugs should be used at the lowest effective dosage and for periods as short as possible. For the non-pharmacological management of pain in these patients, a multidisciplinary team including hematologists, orthopedic surgeons, rheumatologists, and physiotherapists is warranted.

R. Gualtierotti · F. Peyvandi  
Università degli Studi di Milano, Dipartimento di Fisiopatologia e Medicina dei Trapianti, Via Pace, 9, 20122 Milan, Italy  
e-mail: roberta.gualtierotti@unimi.it

R. Gualtierotti · F. Tafuri · S. Arcudi · F. Peyvandi  
Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Medicina Interna – Emostasi e Trombosi e Centro Emofilia e Trombosi, Milan, Italy

P. L. Solimeno · J. Acquati  
Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Ortopedia e Traumatologia, Milan, Italy

L. Landi  
Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Anestesia e Terapia Intensiva Donna-Bambino, Milan, Italy

**Keywords:** Acute pain; Chronic pain; Hemarthrosis; Hemophilic arthropathy

△ Adis

**Fig. 2**  $d_1$  with highlighted metadata

**Extraction results**

<b>Metadata</b>	<b>References</b>	<b>Full text</b>	<b>NLM</b>
Extracted metadata formatted in HTML form. Please see NLM for full extraction results.			
<b>Article title:</b>			Current and Emerging Approaches for Pain Management in Hemophilic Arthropathy
<b>Author:</b>			Roberta Gualtierotti OL. Landi Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Anestesia e Terapia Intensiva Donna- Bambino, Milan, Italy 1P. L. Solimeno J. Acquati Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Ortopedia e Traumatologia, Milan, Italy 2R. Gualtierotti (&) F. Peyvandi Universita' degli Studi di Milano, Dipartimento di Fisiopatologia e Medicina dei Trapianti, Via Pace, 9, 20122 Milan, Italy 3R. Gualtierotti F. Tafuri S. Arcudi F. Peyvandi Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Medicina Interna - Emostasi e Trombosi e Centro Emofilia e Trombosi, Milan, Italy roberta.gualtierotti@unimi.it
<b>Author:</b>			. Francesco Tafuri . Sara Arcudi . OL. Landi Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Anestesia e Terapia Intensiva Donna- Bambino, Milan, Italy 1P. L. Solimeno J. Acquati Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Ortopedia e Traumatologia, Milan, Italy 2R. Gualtierotti (&) F. Peyvandi Universita' degli Studi di Milano, Dipartimento di Fisiopatologia e Medicina dei Trapianti, Via Pace, 9, 20122 Milan, Italy 3R. Gualtierotti F. Tafuri S. Arcudi F. Peyvandi Fondazione IRCCS Ca' Granda Ospedale Maggiore Policlinico, Medicina Interna - Emostasi e Trombosi e Centro Emofilia e Trombosi, Milan, Italy
<b>Publisher:</b>			
<b>Journal title:</b>			Pain Ther
<b>Journal ISSN:</b>			
<b>Volume:</b>			11
<b>Issue:</b>			
<b>Pages:</b>			1-15
<b>Abstract:</b>			Introduction: Hemophilia is an inherited bleeding hematological disorder characterized by the partial or complete deficiency of clotting factor VIII or IX. Hemophilic arthropathy is the consequence of repeated joint bleeding (hemarthrosis) and its management is based on the prevention of acute bleeding through the administration of the deficient clotting factor concentrate or non-factor therapies. In addition, the management of acute and chronic pain is pivotal in hemophilic arthropathy in order to restore function and allow rehabilitation of the joint.
<b>Keywords:</b>			Acute pain; Chronic Hemarthrosis; Hemophilic arthropathy
<b>DOI:</b>			
<b>URN:</b>			
<b>Publication date:</b>			2022
<b>Received date:</b>			2021-8-7
<b>Revised date:</b>			
<b>Accepted date:</b>			2021-12-7

**Fig. 3** Metadata extracted by CERMINE<sup>10</sup> from *d*<sub>1</sub>

of a given metadata extraction system. In an increasingly data-driven world, having quality data or metadata means helping to support the veracity of information.

A future line of research could be the extension of metadata verification and modelling to other types of document sources, with the possibility of using and comparing different metadata extraction systems.

<sup>10</sup> <http://cermine.ceon.pl/cermine/task.html;jsessionid=D89E6E61707120A06D774DB8E5B2F837?task=2743819617862135913>.

Finally, another line of research could be the possibility of experimenting with new modal logics to be applied to data science and addressing “difficult questions about the *modal* basis of scientific modelling, where the central issues concern the nature and justification of the modal content of claims made on the basis of models” [12].

## References

1. Belnap N (1977) A useful four-valued logic. In: Dunn JM, Epstein G (eds) Modern uses of multiple-valued logic. Springer, Berlin, pp 5–37. [https://doi.org/10.1007/978-94-010-1161-7\\_2](https://doi.org/10.1007/978-94-010-1161-7_2)
2. Cuconato S (2021) Epistemic logic for metadata modelling from scientific papers on COVID-19. *Sci Philos J Epistemol Sci Philos* 9(2):83–96. <http://dx.doi.org/10.23756/sp.v9i2.652>
3. Cuconato S (2022) A logical-metaontological approach to the problem of (meta)data veracity in systems for automatic extraction of metadata from scientific-legal articles. *Sci Philos J Epistemol Sci Philos* 10(2):168–187. <http://dx.doi.org/10.23756/sp.v10i2.784>
4. Dunn J (1976) Intuitive semantics for first-degree entailments and ‘coupled trees.’ *Philos Stud* 29(3):149–168
5. Gualtierotti R, Tafuri F, Arcudi S, Solimeno PL, Acquati J, Landi L, Peyvandi F (2022) Current and emerging approaches for pain management in hemophilic arthropathy. *Pain Ther* 11(1):1–15. <https://doi.org/10.1007/s40122-021-00345-x>
6. Hintikka J (1962) Knowledge and belief: an introduction to the logic of the two notions. In: Hendriks VF, Symons J (eds) Texts in philosophy, 2nd ed, 1. College Publications, London
7. Hosni H, Vulpiani A (2018) Data science and the art of modelling. *Lett Mat Int* 6:121–129. <https://doi.org/10.1007/s40329-018-0225-5>
8. Pomerantz J (2015) Metadata. MIT Press Ltd
9. Priest G (2008) An introduction to non-classical logic: from if to Is, 2nd edn. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9780511801174>
10. Quine WV (1969) Ontological relativity and other essays. Columbia University Press, New York
11. Santos YD (2020) A four-valued dynamic epistemic logic. *J Logic Lang Inform* 29:451–489. <https://doi.org/10.1007/s10849-020-09313-8>
12. Tahko TE (2023) The modal basis of scientific modelling. *Synthese* 201(75):1–16. <https://doi.org/10.1007/s11229-023-04063-z>
13. Tkaczyk D, Szostek P, Jan Dendek P, Fedoryszak M, Bolikowski Ł (2014) CERMINE—automatic extraction of metadata and references from scientific literature. In: Conference: 2014 11th IAPR International workshop on document analysis systems
14. Tkaczyk D, Szostek P, Jan Dendek P, Fedoryszak M, Bolikowski Ł (2015) CERMINE—automatic extraction of metadata and references from scientific literature. *Int J Doc Anal Recogn* (IJDAR). <https://doi.org/10.1109/DAS.2014.63>
15. van Ditmarsch H, Halpern J, van Der Hoek W, Kooi B (2015) Handbook of epistemic logic. College Publications
16. Weisberg M (2013) Simulation and similarity: using models to understand the world. Oxford University Press

# Enhancing IoT Network Security with Light Gradient Boosting Machine and Gravitational Interaction Optimization for Malicious Access Detection



Geetanjali Bhoi, Bighnaraj Naik, and Etuari Oram

**Abstract** As the adoption of IoT devices and networks increases, the effective functioning of the IoT infrastructure requires secure hardware, and device connectivity, and software. To ensure the security of IoT devices and networks, it is essential to analyze data and scan network traffic from these connected IoT devices. This paper proposes an ensemble learning model that uses gravitational interaction optimization to optimize hyperparameters for the detection of malicious traffic in IoT networks. The hyperparameters that affect model accuracy and complexity, such as the number of estimators, number of leaves, regularization term, learning rate, bin subsample size, have been optimized. Several performance metrics, including recall, precision, F-Beta scores, and F1-scores, are used to compare the performance of the proposed approaches. Compared to particle swarm optimization, this hyperparameter optimization approach was much faster in terms of convergence rate and accuracy.

**Keywords** Gravitational interaction optimization · Light gradient boosting machine · Ensemble learning · IoT security

---

G. Bhoi (✉) · B. Naik · E. Oram

Department of Computer Application, Veer Surendra Sai University of Technology, Burla, Odisha 768018, India

e-mail: [gbhoi\\_phdca@vssut.ac.in](mailto:gbhoi_phdca@vssut.ac.in)

B. Naik

e-mail: [bnaik\\_mca@vssut.ac.in](mailto:bnaik_mca@vssut.ac.in)

E. Oram

e-mail: [eoram\\_mca@vssut.ac.in](mailto:eoram_mca@vssut.ac.in)

## 1 Introduction

Embedded technology is one of the important characteristics of the IoT that enables efficient access to inter-state communication and the external world. [1]. The IoT's ubiquitous and scalable features enable a smart living style in smart networks. However, various malicious attacks are also a growing concern. In the twenty-first century, intelligent area networks and innovative network application domains surround us, creating a complex environment for the IoT [2]. New technologies evolve constantly and are often associated with anonymous security threats [2]. Smart networks and devices are not risk-free environments or safe defense systems, and security issues are a growing concern for many companies. Attackers use various methods, including physical attacks, software and hardware attacks, and others such as DDoS, ransomware, botnets, spyware, and jamming, to compromise the security of these systems. Dealing with security threats is challenging due to the various ways vulnerabilities can spread. New detection systems and consequences must be constantly developed to keep up. Intruders have different targets, including mimicry and malicious access, with the primary goal being to elude person or system resources [3]. Congestion occurs when traffic flow exceeds a certain level. Attackers may also use jamming to disrupt signals, obstructing devices' broadcast communication, lowering the energy level of the channel, and subsequently affecting the processing system and access control. Botnets combine numerous strategies to gain complete control over a targeted system and disperse malicious data. Spoofing is a type of cyber-attack in which a perpetrator poses as a trustworthy source to obtain sensitive data, such as identification and financial assets, through websites or phone conversations.

DDoS assaults can occasionally cause the system that handles normal traffic to lag, as they attempt to overload the functioning target system by sending numerous requests periodically. Advanced persistent threats are major targets of cyberattacks. Ransomware is a sophisticated security threat that affects IoT and smart homes, as it targets both hardware and software. In addition to viruses, other malicious tools such as Trojans, adware, spyware, and worms can also harm web applications and lead to the loss of personal data. In this era of cutting-edge technologies and scalable environments, IoT devices and apps unintentionally create many security challenges. If massive threats are expected to spread to Internet-connected devices, security authorities, and resource managers need to be notified promptly. However, manufacturers of IoT devices offer no guarantees about the security measures they will implement [4]. IoT security is an essential component of the IoT network infrastructure. The scalable, dispersed, interconnected, and dependable foundation of the IoT makes attacks viable. Security protection includes message encryption, access control, application and network assurance, and other supporting techniques. However, managing the massive operating systems has become an inadequate and tiresome undertaking. The diverse array of applications available for IoT systems presents a challenge in identifying an effective strategy for safeguarding the system itself. Therefore, IoT systems themselves need to be protected. Preprocessing techniques may be problematic or hinder the working process due to the increasing amount of data produced by linked

devices. Applying machine learning (ML) approaches through IoT can advance the study, regardless of whether it is for data-driven infrastructure challenges for people or for groups of connected devices. [5]. The use of ML-based models may help robots and smart devices to collect artificial data and relevant functional information more effectively.

In this study, a light gradient boosting machine-based model is proposed for identifying malicious access in IoT networks, and its hyperparameters are optimized using gravitational interaction optimization (GIO). The remaining sections of this article are organized as follows: Sect. 2 provides a literature survey, Sect. 3 describes the proposed model, Sect. 4 presents the results analysis, and Sect. 5 concludes the study.

## 2 Literature Survey

As a result of the use of ML methodologies in recent years, Intrusion Detection System (IDS) has evolved into superior fashions providing advanced security due to the alerting of potential threats. Service issues, data privacy, and other anonymous activities are monitored with the aid of software application like IDS. Since the initial stage of the development [6], the overall performance in anomalies detection and safety risk of IDS has regularly advanced [7–12].

The paper [13] proposes a swarm-based fuzzy clustering approach to detect intrusive behaviors. It highlights the need for effective intrusion detection systems and presents the proposed approach, which uses a combination of swarm optimization and fuzzy clustering to identify anomalous behaviors. The approach is evaluated using a dataset and compared to other existing approaches. Mishra et al. [14] introduces a Stacking Ensemble Meta-Learning approach for an IoT security framework. It discusses the challenges of IoT security and presents the proposed approach, which combines multiple machine learning models through a stacking ensemble to improve accuracy in detecting IoT attacks. The approach is evaluated using various datasets and compared to other existing approaches. Another paper [15] presents a novel approach for detecting phishing webpages using the LGBM algorithm. It focuses on identifying phisher website features of mimic URLs and proposes a model based on LGBM to distinguish phishing pages from legitimate ones. The approach is evaluated using a dataset of real-world phishing URLs and compared with other state-of-the-art approaches. The results show that the proposed approach outperforms other methods and has the potential for real-world applications. A LGBM-based model with optimized hyperparameters has been proposed [16] for identifying malicious access in IoT networks. It addresses the challenges of IoT security and presents the proposed approach, which uses LGBM and genetic algorithm-based hyperparameter optimization to improve accuracy in detecting anomalous behavior. The approach is evaluated using a dataset and compared with other state-of-the-art approaches, demonstrating superior performance. Rastegari et al. [17] proposed an approach to

evolving statistical rulesets for network intrusion detection. It highlights the limitations of traditional intrusion detection systems and presents the proposed approach, which uses a genetic algorithm to optimize statistical rulesets for identifying anomalous behavior. The approach is evaluated using a dataset and compared to other existing approaches.

Although the performance of IDS systems has improved somewhat, certain inevitable parameters, such as increased network traffic volumes, selecting optimal decision limits, and imbalanced data, can still cause the system to underperform. While many cutting-edge techniques have been developed to enhance IDS performance, there is still room for scientific and efficient improvement. Several writers have attempted to demonstrate the accuracy and effectiveness of their work, but have fallen short of their goals. In such situations, an Ensemble Learning (EL) model may be a better option. This meta-based technique combines various machine learning models to improve prediction performance more precisely and effectively. EL outperforms other machine learning models in two performance metrics: precise error and less overfitting. In comparison to other models, several well-known dataset sources, such as Kaggle, KDD2009, and Netflix, are commonly used in competitions. In performance comparisons, this method outperforms every other single model.

### 3 Proposed Approach

In this work, an LGBM-based model [18] with GIO [19] has been proposed for IoT environment anomaly detection. We took into account an IoT security data collection made up of service accesses in an IoT network with anomalous access instances. A simulated IoT dataset [20, 21] with 357,952 instances is used to evaluate the proposed model normal type 347,935 and anomaly type 10,017). There are 13 characteristics total, with ‘Normality’ being the final one. In this study, a resampled data collection of 186,712 instances 130698 of the normal type and 56,014 of the anomalous type are employed. This is due to memory and runtime limitations in the implementation and execution of the proposed technique. This dataset can be represented as  $D = \{\{x_1, y_1\}, \{x_2, y_2\} \dots \{x_i, y_i\} \dots \{x_n, y_n\}\}$ , where  $x_i$  is the  $i$ th service access with different characteristics like ‘service ID’, ‘access node type’ etc., and  $y_i$  is the ‘normality’ (either normal/malicious). This suggested method can be viewed as process of looking for the best value of LGBM’ hyperparameters such as no of estimators (ne), number of leaves (nl),  $l_1$ -regularization ( $\alpha$ ),  $l_2$ -regularization ( $\wedge$ ), learning rate (lr), and subsample for bin (sb). The initial population ( $B$ ) of the search is based on a GIO metaheuristic optimization with  $n$  number of hyperparameter specified  $B = \{B_i\}_{i=1}^n$ . Here each  $B_i = \{ne_i, nl_i, \alpha_i, \wedge_i, lr_i, sb_i\}$  stands for a collection of hyperparameter values that were generated at random and selected from a particular range:  $ne_i \in [1, 15]$ ,  $nl_i \in [2, 10]$ ,  $\alpha_i \in [0, 1]$ ,  $\wedge_i \in [0, 1]$ ,  $lr_i \in [0, 1]$ ,  $sb_i \in [1000, 50000]$ . The goal in this case is to identify the best hyperparameter set  $B_i^* = \{ne_i, nl_i, \alpha_i, \wedge_i, lr_i, sb_i\}$  in the search space that is the LGBM model’s optimal parameter set for the desired task. In this work, GIO has been used for finding LGBM

model's optimal parameter set  $B_i^*$  in hyperparameter search space  $B = \{B_1, B_2, \dots\}$  for identification of malicious attack in IoT network. Therefore, this task may be seen as an optimization problem with objective of getting optimal hyperparameter set  $B_i^*$  that maximize the target objective function (Eq. 1).

$$B_i^* \leftarrow \max_{B_i} (f_{LGBM}(D, B_i)) \quad (1)$$

In Eq. 1,  $f()$  is the objective function (f1-score) of the LGBM model obtained for pretrained LGBM model (trained on  $D$ ) with hyperparameter  $B_i$ .

$$F_{ij} = \frac{M(f(B_i)).M(f(B_j))}{|B_i - B_j|^2} \hat{B}_{ij} \quad (2)$$

---

**Algorithm 1:** GIO based hyperparameters optimization of *LGBM*


---

**Begin**

Set the GIO parameters: Gravitational constant  $G$ , Cognitive and gravitational interaction constants  $C_1$ , and  $C_2$

iter=1

**While** (1)

Calculate fitness of each  $B_i$  in  $B$ :

**For** each  $B_i$  in  $B$

$$f_i \leftarrow LGBM(B_i, D)$$

$$f = f \cup f_i$$

Calculate the force  $F_{i,j}$  acts on mass  $M_i$  from  $M_j$ :

**For**  $i = 1$  to  $n$

**For**  $j = 1$  to  $n$

Calculate the force  $F_{i,j}$  using Eq.2.

**For** each in  $B_i$  in  $B$ , Calculate the total force  $F_i$  (Eq.4) acts on mass  $M_i$

**For** each in  $B_i$  in  $B$ , Find location of particle  $B_k$  using Eq.5

Find next velocity and net position of each  $B_i$ :

**For**  $i = 1$  to  $n$

$$V_i^{next} = \chi(V + C_1.rand(0,1).(B^b - B) + C_2.rand(0,1).B_k)$$

$$B_i^{next} = B_i + V_i^{next}$$

Update:  $V = V^{next}$ ,  $B = B^{next}$

**If** (iter==Max OR improvement in fitness if less than a threshold value)

    Exit from While

**Else** iter = iter + 1

Return best  $B_i$  from  $B$

---

In Eq. 2,  $B_i$  is the  $i$ th body position and the  $j$ th contributing body is  $B_j$ .  $B_{ij}$  is a unit vector between the entities  $B_i$  and  $B_j$  and  $|B_i - B_j|$  represents the Euclidean distance.  $f(B_i)$  is the  $B_i$  field's fitness, and  $M$  is the mapping function. Let  $f_i$  is the fitness value of the position of  $B_i$ . The mapping function is calculated using Eq. 3.

$$M(f(B_i)) = \left( \frac{f(B_i) - \min f(B)}{\max f(B) - \min f(B)} (1 - \text{mapMin}) + \text{mapMin} \right)^2 \quad (3)$$

Here,  $\min f(B)$  is the minimum fitness value all the positions,  $\max f(B)$  is the maximum fitness value of all the positions so far. By rescaling the value  $f(B_i)$  to a mass within the  $(1 - \text{mapMin})$  interval using a constant  $\text{mapMin}$  that is close to zero with a small positive value, the resulting mass is squared to determine the optimal and suboptimal shape. The approach is characterized by full interaction, meaning that everybody  $B_i$  interacts with each other body  $B_j$  through their individual masses, which in turn affects their motion based on the resultant force calculated by Eq. 4.

$$F_i = \sum_{j=1}^n \frac{M(f(B_i)).M(f(B_j^b))}{|B_i - B_j^b|^2} B_i \hat{B}_j^b \quad (4)$$

Here in Eq. 4,  $F_i$  (resultant force) between  $M(B_i)$  and  $M(B_j^b)$  can be found here. Additionally,  $|B_i - B_j^b|$  is the Euclidean distance between the current position of body  $B_i$  and its best position of body  $B_j$  so far. If we only calculate the force between masses  $M(B_i)$  and  $M(B_j^b)$  is calculated only if the distance between them is smaller, then is the single vector controlling the force  $B_i \hat{B}_j^b$ ,  $j$  is the single vector controlling the force. To find the position of body where  $B_k$ , where  $M(f(B_k)) = 1$ ,  $B_k$  is calculated by Eq. 5.

$$B_k = \sqrt{\frac{M(f(B_i))}{|F_i|} \times \hat{F}_i} \quad (5)$$

The inertial constraint  $\chi$  prevents the solution from exploring other solutions outside the search space that was calculated by Eq. 6.

$$\chi = \frac{2K}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (6)$$

If  $\phi$  is greater than 4, then our algorithm will choose a value for  $\kappa$  that is between 0 and 1 as in [22]. In this case,  $K$  is set to 2.01.

---

**Algorithm 2:** Evaluate the goodness of  $B_i$  in *LGBM* using dataset  $D$

$$f_i = LGBM(B_i, D)$$


---

Finding the mutually exclusive features in  $D = \{x_i, y_i\}_{i=1}^n$  and merging them using EFB (exclusive feature bundling).

$$\text{Create initial model } (M_0(D) = \operatorname{argmin}_{\hat{y}_i} \left( \sum_{i=1}^n \ell\left(y_i, \hat{y}_i\right) \right))$$

For  $i = 1$  to  $K$

$$\text{Calculate the absolute gradients } r'_i \text{ from } M_i(D) = M_{i-1}(D)$$

Generate  $S_1$  and  $S_2$  by Gradient-based One-Side Sampling (GOSS) using  $r'_i$

Combine the subsets  $S_1$  and  $S_2$  to create a dataset  $D^* = S_1 \cup S_2$

Calculation of estimated variance information gain

Derive a new decision tree  $M_i(D)^*$  from  $D^*$

$$M_i(D) = M_{i-1}(D) + M_i(D)^*$$


---

## 4 Result Analysis and Discussion

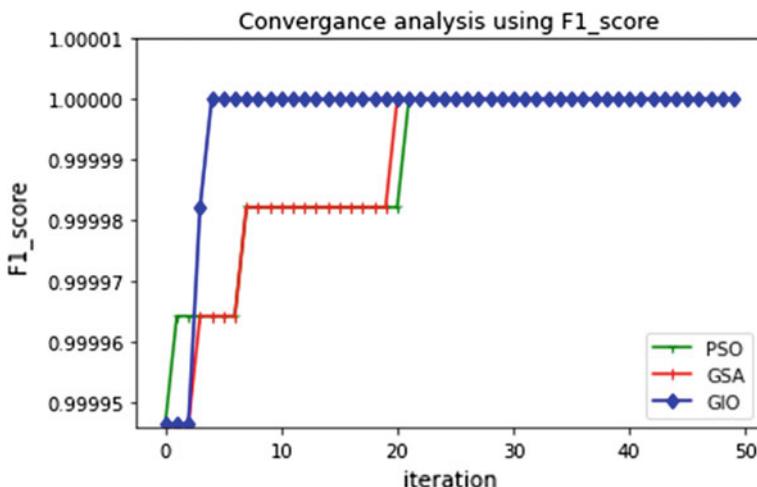
This section compares different machine learning models and identifies the model that can achieve the best performance. Specifically, we compare the performance of Linear Discriminant Analysis (LDA), Decision Tree (DT), Naïve Bayes (NB), Linear Regression (LR), and Multilayer Perceptron (MLP) ML models. We also conduct a comparison of various Ensemble Learning (EL) models, including Bagging, XGBoost, Gradient Boosting Decision Tree (GBT), AdaBoost, and Light Gradient Boosting Machine (LGBM), to see how they perform. All considered models are evaluated and compared. Our analysis reveals that all studied models, including the proposed approach, demonstrate high precision, recall, and F1-score. To evaluate the models, we split the original data into 70% for training and 30% for testing. Table 1 presents the evaluation metrics obtained from the test data for all models.

Table 1 shows a comparison of the performance of the proposed GIO + LGBM model with the other 13 models. The proposed model outperforms all other models, except for PSO + LGBM and GSA + LGBM, which have inferior performance. In terms of performance metrics, the proposed models GIO + LGBM, GSA + LGBM, and PSO + LGBM perform equally well (based on the considered metrics). However, concerning convergence speed, the proposed technique is found superior to PSO + LGBM and GSA + LGBM (shown in Fig. 1).

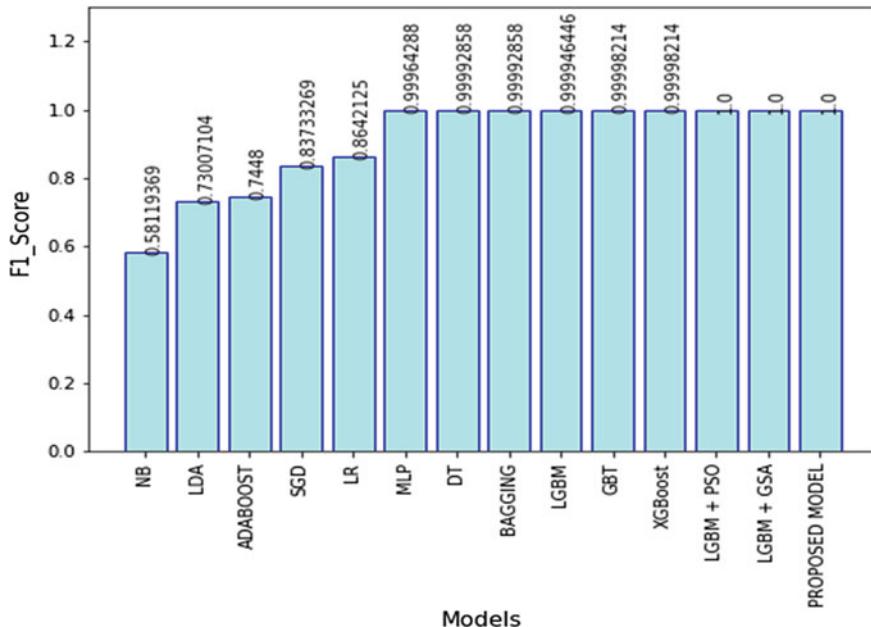
Various optimal hyperparameters sets {ne, nl,  $\alpha$ ,  $\wedge$ , lr, and sb} from PSO, GSA, and GIO are taken into consideration [10,

**Table 1** Comparison of various performance metrics

Prediction models	Performance metrics			
	Recall	Precision	F-beta score	F1-score
LDA	0.75356160	0.77016901	0.73817036	0.73007104
DT	0.99992858	0.99992859	0.99992858	0.99992858
NB	0.66503017	0.69704228	0.56638119	0.58119369
LR	0.87044310	0.86347558	0.86294284	0.86421250
SGD	0.84887706	0.84025176	0.83672339	0.83733269
MLP	0.99964294	0.99964371	0.99964327	0.99964288
AdaBoost	0.7516	0.7724	0.7558	0.7448
Bagging	0.99992858	0.99992860	0.99992859	0.99992858
XGBoost	0.99998214	0.99998214	0.99998214	0.99998214
GBT	0.99998214	0.99998214	0.99998214	0.99998214
LGBM	0.99994644	0.99994646	0.99994645	0.99994644
LGBM + PSO	1.0	1.0	1.0	1.0
LGBM + GSA	1.0	1.0	1.0	1.0
Proposed Model	1.0	1.0	1.0	1.0

**Fig. 1** Convergence analysis (Iteration vs F1\_score)

0.7079175346037112, 13, 0.2977317836668294, 0.02717952920389799, 9162}, {5, 0.9978034920533025, 12, 0.3618380483295082, 0.285327129970158, 4300}, and {7, 0.7072743075458022, 15, 0.36561862813563717, 0.4066382654207831, 13,933} respectively. In Fig. 2, the F1-score for each machine learning model and ensemble learning model has been compared.



**Fig. 2** Assessment of the performance of the models under consideration

## 5 Conclusion

Various machine learning (ML) models have been developed to detect malicious activities in IoT systems, but accuracy remains a major challenge. In this study, we propose an ML approach with optimized hyperparameters that outperforms other approaches. With fewer parameters and the best estimators, our proposed model achieved higher accuracy in identifying security threats in IoT systems. However, ML-based IoT security requires a large and high-quality dataset to train the model. Although there is an abundance of IoT network data available from various sources, the heterogeneity of these data makes it difficult to train each algorithm. Therefore, a crowd-sourcing platform should be created to generate diverse datasets for various security requirements. ML algorithms require datasets that represent all kinds of authentication and attack scenarios. This requires heavy computational resources and is one of the challenges in developing a generalized dataset that captures all attack types. Moreover, ML models must be trained with large datasets before being deployed, which also requires high computational resources.

**Acknowledgements** The funding for this work has been provided by the Department of Science and Technology (DST), Ministry of Science and Technology, Government of India, under the Grant number DST/INSPIREFellowship/2019/IF190611.

## References

1. Varga P et al (2017) Security threats and issues in automation IoT. In: 2017 IEEE 13th International workshop on factory communication systems (WFCS). IEEE. <https://doi.org/10.1109/WFCS.2017.7991968>
2. Bhoi G et al (2022) Gravitational search optimized light gradient boosting machine for identification of malicious access in IoT network. In: Computational intelligence in pattern recognition: proceedings of CIPR 2022. Springer Nature Singapore, Singapore, pp 570–579
3. Lam B, Larose C (2016) How did the internet of things allow the latest attack on the internet?
4. Rouse M (2013) Iot security (internet of things security). Available on: <http://internetofthingsagenda.techtarget.com/definition/IoT-securityInternet-of-Things-security>
5. Hasan M et al (2019) Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things 7:100059. <https://doi.org/10.1016/j.iot.2019.100059>
6. Denning DE (1987) An intrusion-detection model. IEEE Trans Softw Eng SE-13(2):222–232. <https://doi.org/10.1109/TSE.1987.232894>
7. Korkmaz SA, Karatas F (2018) Big data: controlling fraud by using machine learning libraries on spark. Int J Appl Math Comput Sci 6(1):1–5. <https://doi.org/10.18100/ijamc.2018138629>
8. Peng K, Leung VCM, Huang Q (2018) Clustering approach based on mini batch k-means for intrusion detection system over big data. IEEE Access 6:11897–11906. <https://doi.org/10.1109/ACCESS.2018.2810267>
9. Peng K, Leung VCM, Zheng L, Wang S, Huang C, Lin T (2018) Intrusion detection system based on decision tree over big data in fog environment. Wirel Commun Mob Comput 2018(4680867):1–10. <https://doi.org/10.1155/2018/4680867>
10. Belouch M, Hadaj SE, Idhammad M (2018) Performance evaluation of intrusion detection based on machine learning using apache spark. Procedia Comput Sci 2127:1–6. <https://doi.org/10.1016/j.procs.2018.01.091>
11. Vimalkumar K, Radhika N (2017) A big data framework for intrusion detection in smart grids using apache spark. In: 2017 International conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 198–204. <https://doi.org/10.1109/ICACCI.2017.8125840>
12. Dahiyya P, Srivastava DK (2018) Network intrusion detection in big dataset using spark. Procedia Comput Sci 132:253–262. <https://doi.org/10.1016/j.procs.2018.05.169>
13. Mishra D, Naik B (2019) Detecting Intrusive Behaviors using swarm-based fuzzy clustering approach. In: Soft computing in data analytics: proceedings of international conference on SCDA 2018. Springer Singapore
14. Mishra D et al (2021) SEM: Stacking ensemble meta-learning for IOT security framework. Arab J Sci Eng 46(4):3531–3548
15. Oram E et al (2021) Light gradient boosting machine-based phishing webpage detection model using phisher website features of mimic URLs. Pattern Recogn Lett 152:100–106
16. Mishra D et al (2023) Light gradient boosting machine with optimized hyperparameters for identification of malicious access in IoT network. Digit Commun Netw 9(1):125–137
17. Rastegari S, Hingston P, Lam CP (2015) Evolving statistical rulesets for network intrusion detection. Appl Soft Comput 33:348–359. <https://doi.org/10.1016/j.asoc.2015.04.041>
18. Ke G et al (2017) Lightgbm: A highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 30
19. Flores JJ, Rodrigo López, Barrera J (2011) Gravitational interactions optimization. In: Learning and intelligent optimization: 5th International conference, LION 5, Rome, Italy, January 17–21, 2011. Selected Papers 5. Springer Berlin Heidelberg
20. <https://www.net.in.tum.de/fileadmin/bibtex/publications/theses/ba-aubet.pdf>. Thesis: Machine learning-based adaptive anomaly detection in smart spaces. Technische Universität München, Department of Informatics
21. Hasan M et al (2019) Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things 7:100059

22. Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), vol. 1. IEEE

# Anti-lock Braking System Using Monte Carlo Simulations



Ibidun Christiana Obagbuwa, Vincent Mohale Zibi, and Mishi Makade

**Abstract** Motor vehicle safety is a major problem in the current era since the fatality rate of road users remains high, and it is the second leading cause of unnatural death globally. Car manufacturers have invested a lot in the previous years, regarding car security systems to curb the mortality rate among road users. The car security system consists of the active security system and the passive security system. In our study, we will be focused on the anti-lock braking system, which is a vital active security feature, and how it is incorporated in modern cars to prevent road accidents. A vehicle safety feature called the anti-lock braking system (ABS) makes it possible for a car's wheels to stay in tractive contact with the ground when braking. In response to the driver's inputs, the ABS functions to stop the wheels from locking up and sliding uncontrollably. A few factors impact the dynamics of the anti-lock braking system controller. The controller must use a controlled torque to maintain the desired wheel slip ratio. The speed and spin of the vehicle are used to compute slip ratio. This article entails how we are going to make use of MATLAB Simulink software to simulate the anti-lock braking systems by performing Monte Carlo simulations to improve the stopping distance, wheel speed, and slip of a vehicle.

**Keywords** Anti-lock braking system · Simulink · MATLAB · Monte Carlo simulation · Sensitivity analysis

---

I. C. Obagbuwa (✉) · V. M. Zibi · M. Makade

Department of Computer Science and Information Technology, Sol Plaatje University, Kimberley, South Africa

e-mail: [Ibidun.obagbuwa@spu.ac.za](mailto:Ibidun.obagbuwa@spu.ac.za)

V. M. Zibi

e-mail: [201902877@spu.ac.za](mailto:201902877@spu.ac.za)

M. Makade

e-mail: [201902668@spu.ac.za](mailto:201902668@spu.ac.za)

## 1 Introduction

The car has become the most significant method of transportation today, thanks to the fast growth of the modern automobile industry during the last century. According to the WHO, more than a million people die each year as a result of motor vehicle accidents worldwide, and motor vehicles are the main cause of mortality for those under the age of 29 [1]. Additionally, it has been shown that the second-largest cause of unnatural fatalities, after suicide, is motor vehicle accidents, which are the main cause of unexpected death. Traditional brakes are simple, the driver pushes the brakes, brake pads apply pressure, and the car slows down. But on slippery surfaces, if the brakes are clamped hard the wheels stop turning and begin to slide on the surface. Sliding is very dangerous as it may cause the vehicle to skid unpredictably out of control and cause accidents [2]. Penny explains that faulty brakes are second on the list of causes of fatal auto accidents, accounting for 15% of all incidents [3]. The anti-lock braking system (ABS) is a vital active safety feature. It can improve driving stability, shorten braking distances, and, to some extent, prevent accidents from occurring as depicted in Fig. 1. Anti-lock braking technology has been improving since the 1990s, and ABS systems have now become standard equipment in automobiles. ABS regulates brake line pressure independently of pedal effort to return the wheel speed for effective braking performance, and a slip level range is necessary [2]. Wheel speed sensors, a hydraulic modulator, and a brake controller are all included. The electronic control unit makes up an anti-lock system. The ABS has a feedback control system that modifies the braking pressure in response to wheel deceleration and wheel angular velocity to prevent the controlled wheel from locking. The device turns off when the vehicle speed drops below a certain threshold [2, 3]. This study aims to implement an efficient ABS in MATLAB with the help of Monte Carlo simulations. Monte Carlo sensitivity analysis is used to determine the relationship between model parameters and slip, wheel speed, and braking distance.

In the case of an anti-lock braking system (ABS), Monte Carlo simulations can be used to model the system's performance and evaluate its effectiveness under different driving scenarios and parameters. The major motivation for choosing Monte Carlo simulations for an ABS is that they allow for a more realistic and accurate representation of the system's behavior compared to simpler analytical or numerical models. One of the advantages of Monte Carlo simulations is that they can incorporate random variations and uncertainties in the system parameters and input conditions, which can have a significant impact on the system's performance. By running multiple simulations with different parameter values and input conditions, Monte Carlo simulations can provide a more comprehensive analysis of the system's behavior and help identify potential issues or design improvements. There have been previous studies that have used Monte Carlo simulations to analyze the performance of ABS, but the proposed work may have unique aspects or contributions that distinguish it from existing studies. For example, the proposed work may focus on a particular type of vehicle or driving scenario, or it may use different performance metrics or input conditions compared to the previous studies. By comparing the proposed work to



**Fig. 1** Vehicle with or without ABS (<https://onroad.com.au/Blog/What-is-the-difference-between-ABS-vs-No-ABS>)

similar previous literature, the authors can demonstrate the novelty and value of their approach and contribute to the ongoing research in this field.

The rest of the paper proceeds as follows: Sect. 2 presents the literature review, Sect. 3 presents the essentiality of anti-lock braking systems, Sect. 4 explains the Simulink model and mathematical modeling, Sect. 4.6 describes Monte Carlo simulation, Sect. 4 presents the results and discussion. Finally, Sect. 5 concludes the paper with a summary of the main points and outcomes.

## 2 Literature Review

Abro et al. [4] proposed a technique that incorporates dynamics of a vehicle body and system modeling. To regulate slip when excessive slide occurs, they applied a basic Pi controller, a tire model, and an actuator model. They used 3 cases in the model which are 3 different types of surfaces, namely concrete, dry, or asphalt. To make their system give good results, they used the standard Pacejka magic tire formula. Results of this study showed that the ABS controlled the slip for all 3 cases and kept it close to the optimum slip value, therefore preventing skidding. Shewale and Deivanathan [5] used MATLAB to model the ABS model for a single wheel. They used CARSIM simulation software to simulate a complete four-wheel simulation. For the hatchback and van sectors of the automobile, a comprehensive ABS model is created, and the simulation software's output is compared. Additionally, they modeled a wheel devoid of ABS and contrasted the outcomes with ABS. The results revealed that ABS significantly reduces slip and stopping distance during brake. Xiao et al. [2] used MATLAB Simulink software to implement ABS simulation and ABS modeling and control. They applied a fuzzy control door limit control strategy to make the simulation results more accurate. Anti-lock systems are designed to achieve three

goals which are to reduce stopping distances, to enhance braking maneuverability, and to increase the level of stability [6]. These have further expatiated below:

## 2.1 *The Stopping Distance*

The stopping distance of a vehicle is impacted by its mass, initial speed, and braking force, among other factors. When all other factors are constant, increasing the braking force will result in a shorter stopping distance. The friction coefficient of different surfaces varies, reaching a peak to some extent. An anti-lock system can potentially produce the most effective force and minimize the stopping distance by keeping all wheels at their highest point of friction. However, this objective is limited by the necessity of maintaining vehicle stability and steerability [6].

## 2.2 *Stability*

High friction force may not always be preferred, even while it is advantageous in some circumstances, such as when a car is on a p-split surface (ice and asphalt, for example), when one side of the vehicle has much higher braking force than the other. When the brakes are applied fully on both sides, the automobile pulls to the side with the most resistance, increasing the vehicle's unsteadiness and requiring the driver to make superfluous steering corrections to counteract the yaw moment. If the lower friction coefficient increases, and an anti-lock system can prevent both rear wheels from slipping, the lateral force can be sustained but not at its maximum level. One of the objectives of anti-lock braking systems is to help with instability [6].

## 2.3 *Steerability*

A good peak frictional force control system is necessary to provide outstanding lateral forces and, consequently, adequate steerability. In addition to making minor course changes, being able to steer while braking is essential for maneuvering around obstacles [6–8].

Tire characteristics have a significant impact on a vehicle's response to braking and turning. On vehicles with ABS, tire performance is essential. The teeny-tiny tire contact area between the road and the car must provide all braking and steering forces. The generation of forces that enable traction and sideways movement relies on the tire circumference speed being different from the speed of the vehicle in relation to the road surface, which is referred to as slip. Tire braking force and tire braking slip are usually linked. After reaching its maximal value, increased tire slip produces a decrease in the tire-road friction coefficient. ABS must restrict slip to values less than

the maximum value in order to prevent the wheel from locking. Maximum friction is produced by tires with high peak friction points at 10 to 20% sliding. As tire-road friction rises, the optimal slip value drops [6].

## 2.4 Sensors for Wheel Speed

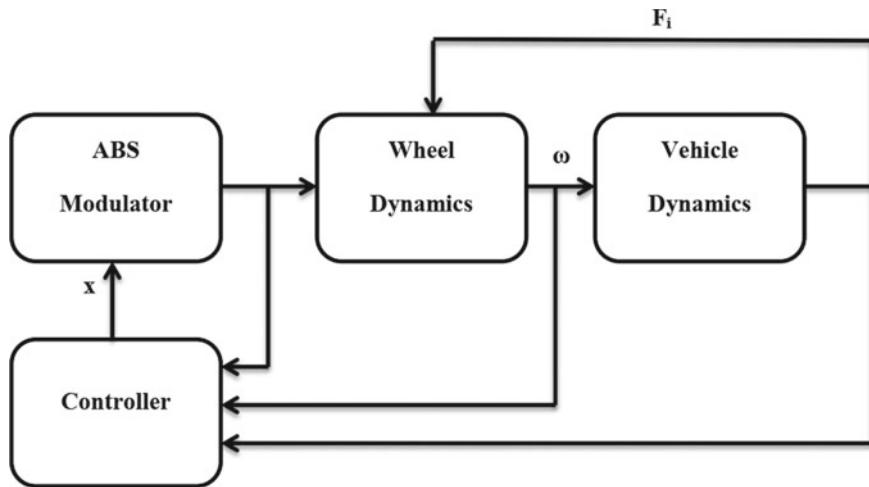
Pulse pickups with toothed wheels are mounted directly on moving drivetrain components or wheel hubs using electromagnetic or Hall effect technology. As a wheel spins, the toothed wheel (pulse ring) initiates an AC voltage at the wheel speed sensor. Voltage frequency corresponds to the rotating speed of the wheel [6–8].

## 2.5 Unit of Electronic Control (ECU)

To ascertain the wheel's rotational speed and acceleration, the electronic control unit acquires, amplifies, and filters sensor inputs. The speed of the vehicle can also be determined by measuring the speed of two diagonally opposed wheels. The estimation of slip at each wheel is carried out by comparing the speed of each wheel to a reference speed. Signals for wheel acceleration and slide are used to alert the electronic control unit about potential locking. Upon receiving an alarm, the micro-computers send a signal to the solenoids of the pressure control valve, which regulate the brake pressure in the wheel brake cylinders through the pressure modulator. To identify issues related to ABS, the electronic control unit utilizes various components such as wheel speed sensors, the pressure control valves, wire harness, and the electronic control unit itself. The troublesome system or ABS is completely switched off when the electronic control unit detects a defect or malfunction [6–8].

## 2.6 A Modulator of Hydraulic Pressure

A hydraulic pressure modulator is an electro-hydraulic device that controls solenoid valves in the hydraulic braking system in order to modulate the pressure of wheel brakes. A brake master cylinder is connected to the wheel brake cylinders using hydraulic fluid. A hydraulic modulator is put in the engine compartment in order to shorten the length of the lines connecting the wheel brake cylinder and brake master cylinder. Based on the device's design, a pump, motor assembly, accumulator, and reservoir may be included [6–8]. Figure 2 illustrates the relationship between modulator, dynamics, and controller. When a wheel locks up when braking, sensors are used by the vehicle dynamics system to identify it. The system then modifies the brake pressure to avoid lock-up and keep the vehicle under control.



**Fig. 2** ABS scheme [6]

### 3 Methodology

The simulation study steps which are shown in Fig. 3 were followed for this work.

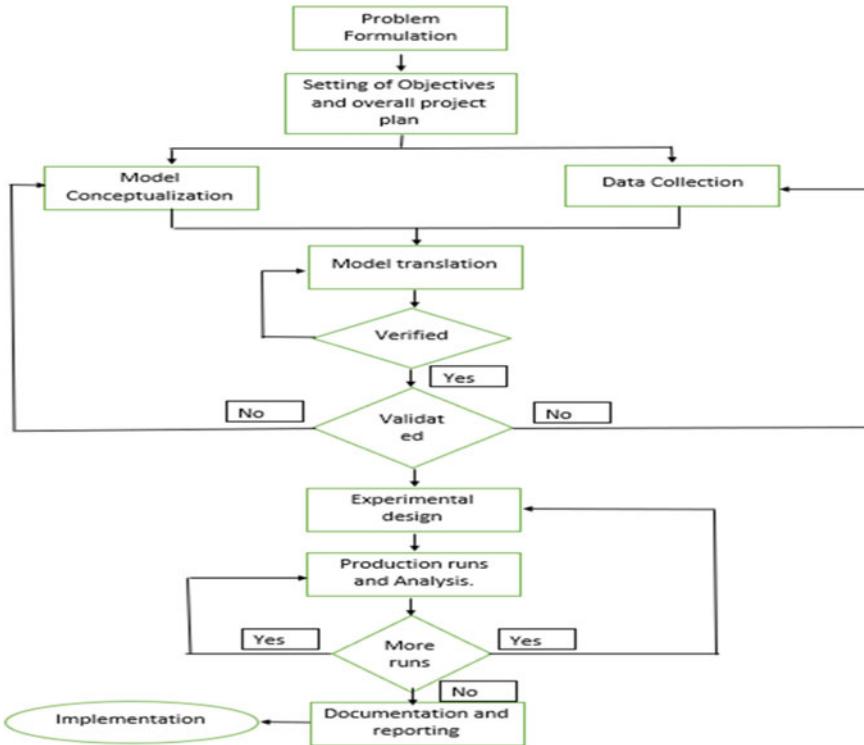
## 4 Modeling and Simulation, Results, and Discussion

### 4.1 Problem Formulation

The relationship between the frictional coefficient and the wheel slip ratio provides an explanation for the ABS. It retains steering stability and stability, while producing shorter stopping distances than a locked wheel stop. The friction coefficient is determined by a variety of factors, including the condition of the surface of the road such as

1. The vehicle speed,
2. The tire brand,
3. The slip ratio of the tire to the road, as well as
4. Tire side-slip angle

For a certain wheel slip ratio, the effective coefficient of friction between the tire and the road has an optimum value. The cost varies according to the kind of road.



**Fig. 3** Simulation study steps [9]

## 4.2 Project Plan

1. This work aims to study the braking systems of cars without ABS and cars with ABS.
2. To compare the two models according to stopping distance, slip, wheel speed, and vehicle speed.
3. To conduct Monte Carlo sensitivity analysis on ABS model to get an insight of how the parameters in the models affect the output of the models. MATLAB Simulink design optimization toolbox will be used to conduct Monte Carlo sensitivity analysis.
4. To use MATLAB Simulink to simulate the model.
5. To collect data from the model and further analyze it in Python.

### 4.3 Model Conceptualization

The ability to pick and modify the fundamental premises that characterize the system, abstract the important components of the problem, build and iterate the model until a reasonable approximation result is obtained all enhance modeling methodologies [9]. It is too complex in control system design to have a full vehicle model with all a vehicle's features. For the controller design, we used a simplified model that included all the elements of a vehicle's system [2]. The single-wheel depiction is shown in Fig. 4.

The degrees of freedom for this model are the automobile's longitudinal velocity and the wheel's rotating speed. [2]. The vehicle model's motions are governed by the Eqs. 1 and 2:

For balancing braking forces in the longitudinal direction (vehicle)

$$ma_x = -\mu F_N \Rightarrow m \frac{dv_x}{dt} = -\mu F_N \quad (1)$$

Adding torque at the wheel's center (wheel):

$$J_{\infty}\alpha_{\infty} = \mu RF_N - T_b \Rightarrow J_{\infty}\omega = \mu RF_N - T_b \quad (2)$$

A slip ratio is defined for convenience as Eq. 3:

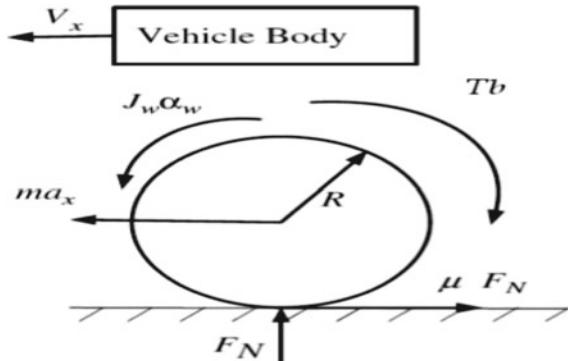
$$\lambda = \frac{V_x - \omega R}{V_x} \quad (3)$$

When we differentiate on both sides in terms of time ( $t$ ), we obtain Eq. 4:

$$\lambda = \frac{V_x(1 - \lambda) - R\omega}{V_x} \quad (4)$$

The following is the terminology used in Eqs. 1–4:

**Fig. 4** Single-wheel depiction [2]



$V_x$  = vehicle's linear velocity,  $a_x$  = linear acceleration of the vehicle,  $T_b$  = braking torque,  $\alpha\omega$  = angular acceleration of the wheel,  $\omega$  = wheel rotational speed,  $\lambda$  = ratio of slippage,  $R$  = tire's radius,  $m$  = the model's mass.

## 4.4 Data Collection

The Simulink models were running during model building, and data was collected and stored as an excel file for further analysis in Python.

## 4.5 Model Translation

Most actual systems develop models that store and compute significant volumes of data. Consequently, the conceptual model must be transformed into a format that computers can understand [9]. Modeling is the initial and very important thing to do in developing a control calculation for the anti-lock braking system. An ABS model has been created in MATLAB Simulink, and all mathematical parts were taken into consideration for computing vehicle speed, wheel speed, stopping distance, etc. The Simulink model is shown in Fig. 5.

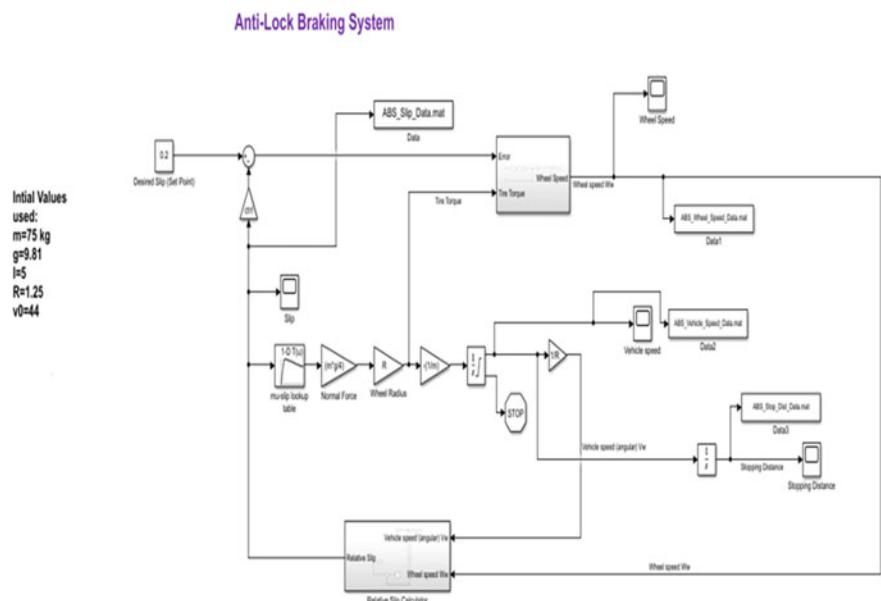
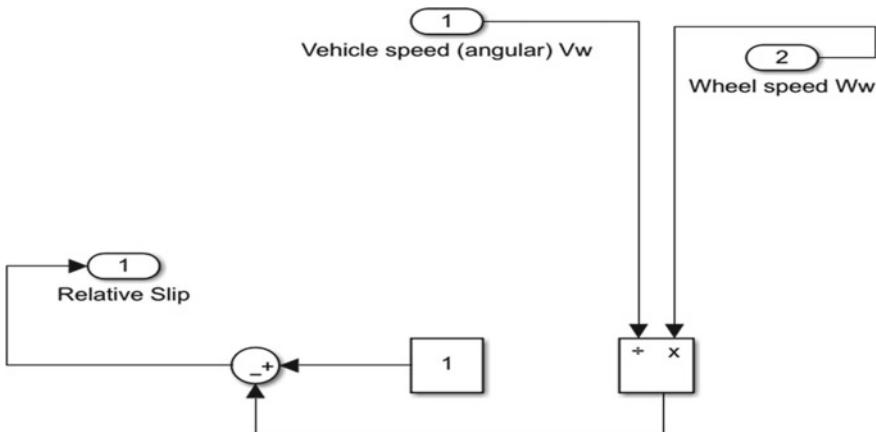


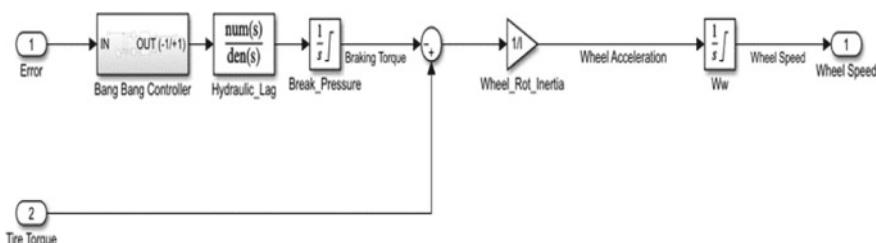
Fig. 5 Single-wheel ABS model developed in MATLAB Simulink

Slip, wheel speed, vehicle speed, and stopping distance are all calculated by the model. The model has a desired slip ratio of 0.2, which is compared to the slip ratio computed by the model. The slip ratio is the proportion of change between automobile speed and wheel speed to car speed. The relative slip sub-system is depicted in Fig. 6. For the computation of car wheel speed, the variation of the two numbers is given to the wheel speed sub-system shown in Fig. 7.

$\mu$  is a coefficient of friction joining the road surface and a tire and is a function of slip which is represented by a  $\mu$ -slip curve. The  $\mu$ -slip curve is generated by inserting MATLAB variables into a Simulink lookup table in a block diagram. The force of friction is applied to the tire circumference which is calculated by multiplying  $\mu$  by the weight of the wheel denoted by  $W$ . To obtain car deceleration, force of friction is divided by vehicle mass. The vehicle deceleration is integrated to get vehicle velocity. Finally, to determine wheel speed, a sub-system is built that employs a bang–bang controller. The actual slip is subtracted from the intended slip, and the difference is sent into the bang–bang controller. Depending on the sign of the mistake, the bang–bang controller returns  $a + 1$  or  $a - 1$ . The on/off value travels



**Fig. 6** Relative slip sub-system



**Fig. 7** Wheel speed sub-system

via a hydraulic lag of first order that imitates delay related to a car system's hydraulic lines.

To obtain the true brake pressure, the filtering rate is integrated. The tire torque is calculated by multiplying the braking pressure by the piston area and the wheel radius. To calculate the torque of acceleration for a road surface on a car wheel, the coefficient of friction was multiplied by the wheel radius. To calculate net torque for the wheel, find the difference between braking torque and accelerating torque. To obtain wheel acceleration, the quotient of rotational inertia and total torque is calculated. The quotient is integrated to get the wheel speed.

## 4.6 Monte Carlo Simulation

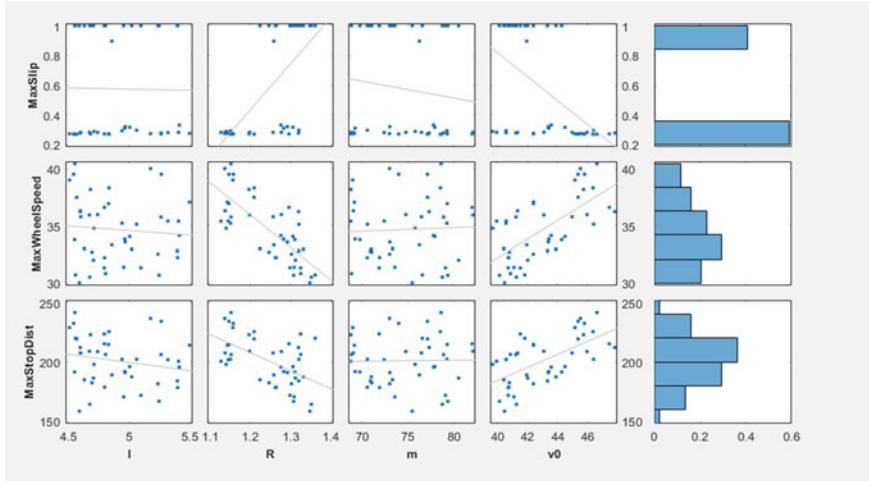
Monte Carlo sensitivity analysis was performed on the model variables to understand which variables affect the output signals of the model and how they affect the output signals of the model. The parameters used in this work for the sensitivity analysis are wheel radius, initial voltage, wheel rotational inertia, and mass of the vehicle. The signals used are slip, wheel speed, and stopping distance. For slip, the Monte Carlo sensitivity analysis was performed to see which variables decrease it so that it gets to the desired slip value of 0.2. For wheel speed and stopping distance, the sensitivity analysis was done to see which variables decrease the two signals.

First, a relationship between the variables and the signals was evaluated. Random samples were generated and used to evaluate the relationship. Figure 8 shows that the wheel radius ( $R$ ) and initial voltage ( $v_0$ ) have a relationship with the stopping distance, wheel speed, and slip. Figure 8 also shows that wheel rotational inertia ( $I$ ) and mass ( $m$ ) of the vehicle have no relationship with slip, stopping distance, and wheel speed.

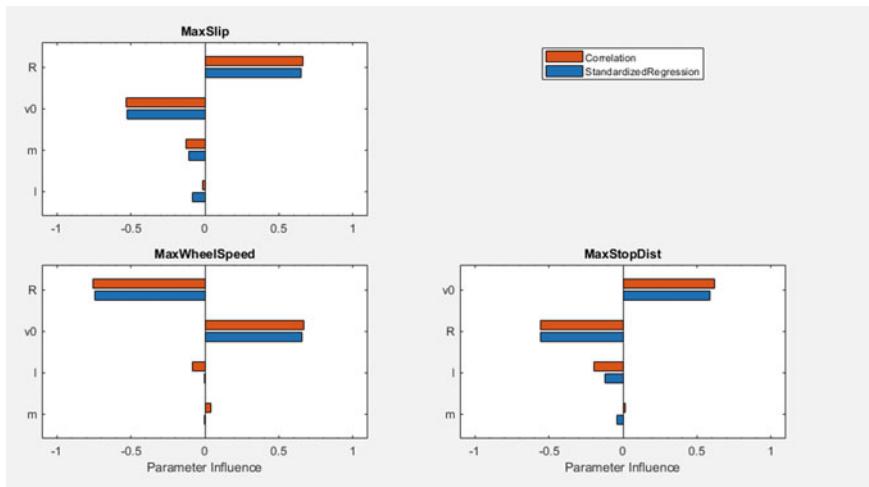
Then, the kind of relationship between the variables and the signals was also evaluated using the same random samples. Figure 9 depicts the relationships.

Figure 9 shows that wheel radius ( $R$ ) has a negative relationship with stopping distance and wheel speed, and initial voltage ( $v_0$ ) has a positive relationship with stopping distance and wheel speed. For slip, wheel radius has a positive relationship with slip, while initial voltage has a negative relationship with the slip. This means that to get low stopping distance and wheel speed, wheel radius must be high and initial velocity must be low and vice versa. The relationship between the initial voltage ( $v_0$ ) and stopping distance in an ABS is complex and indirect, and it is determined by multiple factors that affect the overall performance of the system.

The criteria for stopping distance in anti-lock braking system (ABS) typically depend on various factors, such as vehicle speed, road surface conditions, and driver reaction time. Generally, the primary goal of an ABS is to reduce the stopping distance and maintain directional control of the vehicle during emergency braking.



**Fig. 8** Monte Carlo sensitivity analysis relationship evaluation

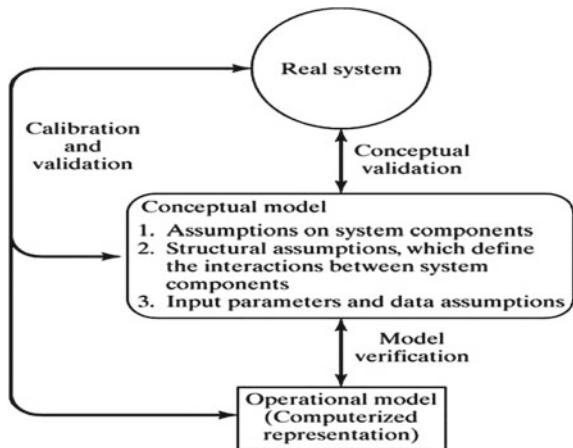


**Fig. 9** Monte Carlo sensitivity analysis type relationship evaluation

#### 4.7 Verification

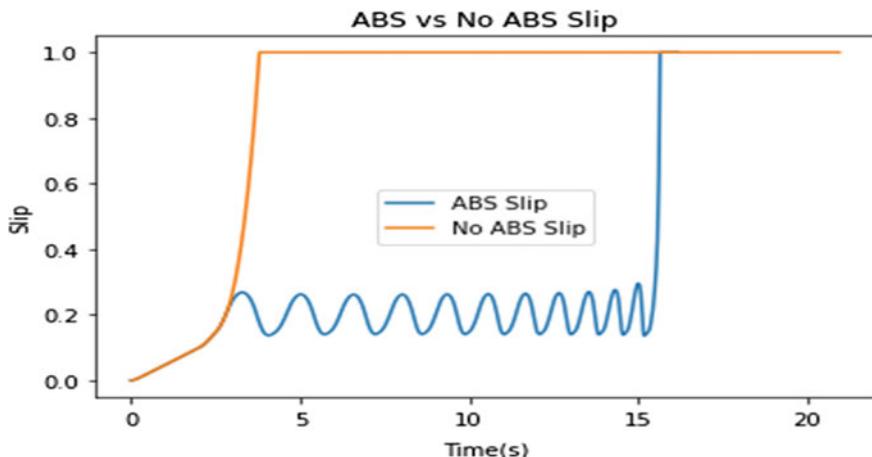
The model underwent a sanity check to guarantee that the conceptual model was accurately converted to operational (computerized model). Figure 10 depicts the model verification process. The results from the MATLAB Simulink model are comprised of the following.

**Fig. 10** Model verification  
[9]



- a. Features of relative slip with and without ABS.
- b. Between-stop distance for ABS and without.
- c. Behavior of wheel speed and vehicle speed with ABS.
- d. Wheel speed behavior and vehicle speed without ABS.

Figure 11 shows the wheel slip ratio with ABS and how it approaches 1 at about 15 s. This implies that the condition of wheel locking is steered clear off, and no immediate locking occurs. When the wheel is not equipped with ABS, a value of 1 is achieved after 5 s, indicating that the wheel locking situation is not prevented.

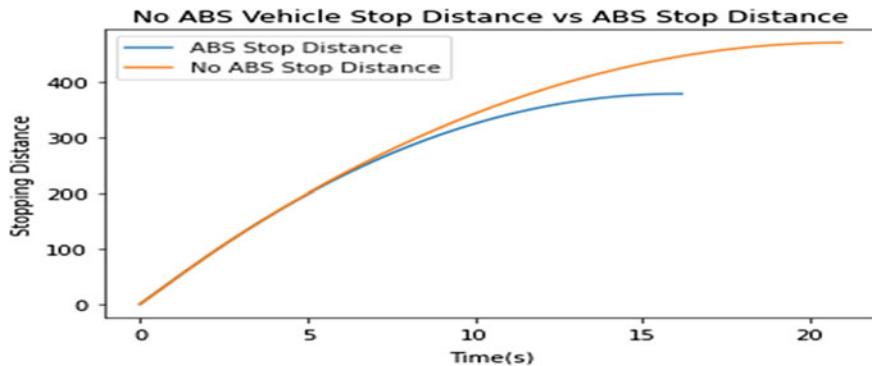


**Fig. 11** Relative slip with and without ABS

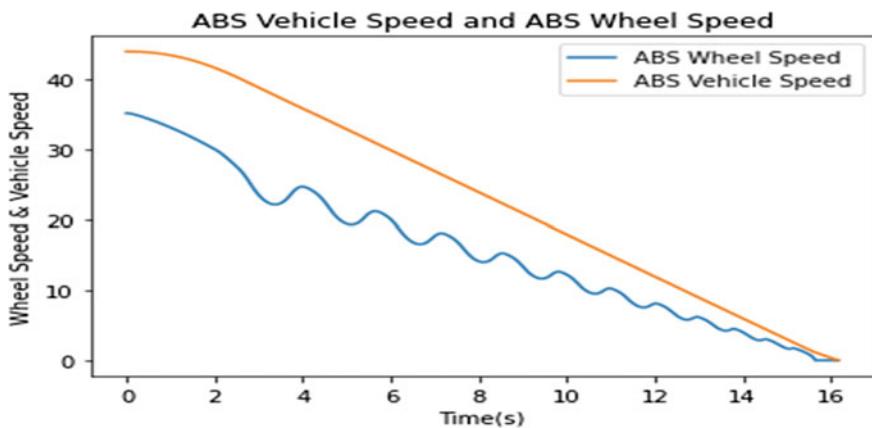
The stopping distance of a car with ABS and a vehicle without ABS is shown in Fig. 12, respectively. The vehicle with ABS reaches the stopping distance faster than the vehicle with no ABS, therefore helping avoid accidents.

Figure 13 demonstrates that both the wheel speed and the vehicle speed decrease, with the wheel speed consistently being lower than the vehicle speed. As a result, the slip ratio will always be positive. The vehicle comes to a stop after 16 s.

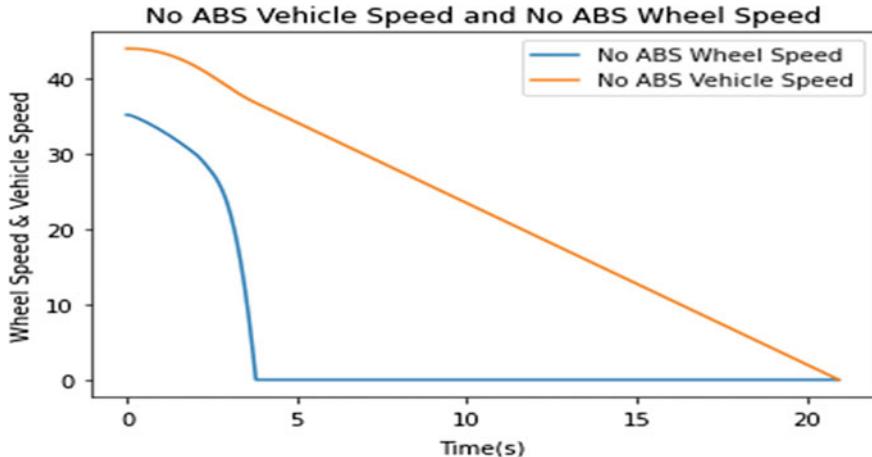
Figure 14 clearly shows that the vehicle and wheel speed behavior without an ABS does not decrease hand in hand, and the wheel speed suddenly reaches 0 within 5 s.



**Fig. 12** Distance between stops with and without ABS



**Fig. 13** Vehicle speed and wheel speed behavior with ABS



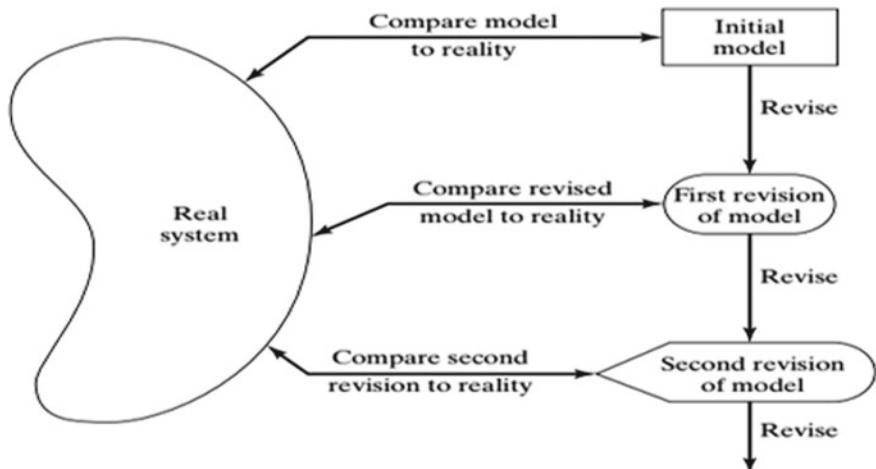
**Fig. 14** Without ABS, the vehicle's speed and wheel speed behavior are uncontrollable

#### 4.8 Validation

Fitting the model is typically how validation is done. This method reduces the disparity between the two by repeatedly comparing the model's behavior to that of the actual system and utilizing the learned information. Repeat this procedure until the model's accuracy is deemed to be adequate [9]. Through the validation process shown in Fig. 15, the model is compared with the actual system to ensure that it can function similarly. Data from the proposed model's history were gathered and examined as the model was being built. The slip ratio of the wheel after using ABS approaches a value of 1 later than a vehicle without ABS, according to model history data. This means that the wheel locking condition is avoided. It also showed that the stopping distance of a vehicle with ABS reaches a stopping distance faster than the vehicle with no ABS. The outcomes also showed that a vehicle with ABS decreases wheel speed and vehicle speed at the same rate when wheel speed is less than vehicle speed. All these insights that we gathered from the dataset resemble the characteristics of a real ABS. Therefore, our model is accurate.

#### 4.9 Experimental Design

The design of experiments requires deciding on alternative simulations. The completed and assessed execution is frequently a determining factor in choosing which alternative to mimic. When constructing a simulated system, choices must be made about the length of the initialization phase, the duration of the simulation run, and the quantity of replicas carried out every run [9]. The following input settings are



**Fig. 15** Model validation process [9]

used to mimic the performance of various vehicle characteristics under straight-line braking to determine the simulation period.

$R = 0.33$ ,  $m = 342$  kg,  $J_w = 1.13 \text{ kgm}^2$ ,  $g = 9.81 \text{ m/s}^2$ , max braking torque = 1200 Nm, initial linear velocity = 27.78 m/s = 100 km/h, initial rotational speed = 84.18 rad/s,  $\lambda d = 0.2$ ,  $K_p = 250$ ,  $K_d = 5$ ,  $K_i = 10$ .

From the time, the brakes were applied until the wheel abruptly stopped, it can be observed that the slip ratio has been fluctuating between 0 and 1. The 45 m stopping distance occurs at 3.6 s, even if at 0.4 s, the wheel speed is zero. This suggests that the wheel was locked before the car came to a stop. That indicates that at 0.42 s into braking, steerability is lost as a result of a wheel locking.

How long the simulation will run will depend on the input parameters. With the above-chosen parameters, the simulation takes 3.6 s to execute.

#### 4.10 Production Runs and Analysis

Production runs and subsequent analysis are used to estimate crucial performance measures for simulated system designs [9]. To measure the performance of the model, apart from MATLAB Simulink, we used Python to measure the accuracy of our system is explained with the aid of some visualizations.

ABS increases the vehicle's braking performance. A proportional controller ( $K_p$ ) would appear to reduce the rising time and the steady-state error, but never totally eliminate it, according to comparisons of the slip ratio vs time graphs of several control systems. Although an integral control ( $K_i$ ) eliminates steady-state error, it may worsen the sudden reaction. A derivative control ( $K_d$ ) improves system

**Table 1** Vehicle's braking performance

Gain response	Rise-time	Overshoot	Settling time
Kp	Reduce	Increase	Minor change
Ki	Reduce	Increase	Eliminate
Kd	Minor change	Reduce	Reduce

stability, reduces overshoot, and improves transient responsiveness. Table 1 outlines the impacts of each of the controllers Kp, Kd, and Ki on a closed-loop system.

## 4.11 More Runs

The analyst assesses if new runs are required based on the analysis of the existing tests and what design those subsequent trials must follow [9]. The model output was collected and analyzed in Python. From the output, we could confirm that the model is accurate and efficient. Therefore, we could conclude that there are no additional runs required.

## 4.12 Documentation and Reporting

An ABS model has been created in MATLAB Simulink, and all mathematical parts were taken into consideration for computing vehicle speed, wheel speed, stopping distance, etc.

### 4.12.1 Relative Slip Sub-system

The slip ratio is the ratio of the difference between the car speed and the wheel speed to the car speed. The slip ratio of 0.2 is achieved by the model. The computed slip ratio is passed through a bang–bang controller which will help with the generation of the slip ratio to make it closer or equal to the desired slip ratio. The wheel speed sub-system is supplied the difference between the slip ratio and the intended slip ratio in order to compute automobile wheel speed.

### 4.12.2 Wheel Speed Sub-system

The wheel speed sub-system is supplied the difference between the slip ratio and the intended slip ratio in order to compute automobile wheel speed. Depending on the sign of the mistake, the bang–bang controller returns  $a + 1$  or  $a - 1$ . The on/off value travels via a hydraulic lag of first order that imitates delay related to a car system's

hydraulic lines. To obtain the true brake pressure, the filtering rate is integrated. Tire torque is calculated by multiplying the braking pressure by the piston area and the wheel radius. To get the torque of acceleration for a road surface on a car wheel, the coefficient of friction was multiplied by the wheel radius. To calculate net torque for the wheel, we used the difference between braking torque and accelerating torque. To obtain wheel acceleration, the quotient of rotational inertia and total torque is calculated. Finally, the quotient is integrated to get the wheel speed.

#### **4.12.3 Bang–Bang Controller Sub-system**

The goal of a controller is to figure out how to generate the appropriate input so that we can get the desired output value. For feedback control controllers, the output of a system is compared to the desired value to see how far the system output is from the desired value. In our case, the bang–bang controller is used to control the value of slip ratio so that it gets to the desired slip value or at least gets closer to the desired slip value. The slip ratio is computed by the model. Then, the difference or error between the desired slip and the computed system slip is calculated. The difference is given to the bang–bang controller. The bang–bang controller generates a value based on the error which indicates whether the calculated slip is closer to the desired slip or not. The value generated by the controller helps the system with generating the appropriate slip ratio.

### **4.13 Implementation**

The success of the implementation phase is determined by how successfully the preceding 11 phases were completed (simulation steps). Successful implementation depends on the continuous participation of the model the successful completion of all process steps. Perhaps the most important point in the whole process is validation (Fig. 15). Invalid models lead to false results and can be dangerous, costly, or both when implemented [9]. Our model was thoroughly verified and validated. A quarter vehicle dynamics model was used to represent the system, and a differential equation of motion was created. For this control job, the slip ratio is employed as a criterion. The friction force and normal response are both functions of the slip ratio, making the equations nonlinear. Three first-order state-space equations were used to represent the second-order differential equations, and MATLAB Simulink block diagrams were used to extract the answers. For a benchmark problem accessible in the literature, the wheel time histories, vehicle stopping distance, and slip factor change are derived. Many central strategies, such as the P-type, PD-type, PI-type, and PID-type, have been used to control the slip ratio.

## 5 Conclusion

This work modeled the braking systems of cars without ABS and cars with ABS, we attempted to clearly understand the application of ABS and how it prevents accidents compared to normal braking systems. The two models were compared according to stopping distance, slip, wheel speed, and vehicle speed. Then, Monte Carlo sensitivity analysis was conducted on the ABS model to get an understanding of how the parameters in the models affect the output of the models. This study made use of MATLAB Simulink to simulate the model. Data collected from the model was further analyzed in Python. MATLAB Simulink design optimization toolbox was used to conduct Monte Carlo sensitivity analysis. According to the findings, the ABS shortens stopping distances and decreases slide to lessen or perhaps avoid accidents. Additionally, it maintains the wheel speed and vehicle speed at a manageable level during braking to enable the driver to maneuver the car while braking, assisting the driver in avoiding obstructions even while braking.

**Acknowledgements** The infrastructural assistance provided for this research by Sol Plaatje University is gratefully acknowledged by the authors.

**Data Availability** The relevant author can provide the data generated for the work upon request.

**Conflicts of Interest** It is stated by the authors that they have no competing interests.

## References

1. World Health Organization (WHO) (2021) Road traffic injuries, 21 June 2021. <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>. Accessed 28 Dec 2021
2. Xiao L, Hongqin L, Jianzhen W (2016) Modeling and simulation of anti-lock braking system based on fuzzy control. IARJSET 3:110–113. <https://doi.org/10.17148/IARJSET.2016.31021>
3. Penny W, Els S (2016) The test and simulation of ABS on rough, non-deformable terrains. J Terramech 67:1–10. <https://doi.org/10.1016/j.jterra.2016.05.001>
4. Abro M, Shaikh S, Asghar A, Mian S, Kalwar I (2018) Design and analysis of anti-lock braking system. In: 2018 3rd International electrical engineering conference (IEEC 2018). IEP Centre, Karachi, Pakistan
5. Gong T, Yan H, Liu P (2014) Modeling and simulation for anti-lock braking system (ABS) of automobiles based on Simulink. Appl Mech Mater 716–717:1504–1507. <https://doi.org/10.4028/www.scientific.net/AMM.716-717.1504>
6. Sharkawy AB (2010) Genetic fuzzy self-tuning PID controllers for antilock braking systems. Eng Appl Artif Intell 23:1041–1052
7. LiveAbout (2021) Non-ABS vs. ABS (Anti-locking brake system) braking systems [online]. Available at: <https://www.liveabout.com/proper-braking-abs-vs-non-abs-3234376>. Accessed 29 Dec 2021
8. Cottingham D (2021) How does anti-lock braking (ABS) work? [online]. Driver Knowledge Test (DKT) Resources. Available at: <https://www.driverknowledgetests.com/resources/how-does-anti-lock-braking-abs-work/>. Accessed 29 Dec 2021
9. Banks J, Carson JS, Nelson BL, Nicol D (2010) Discrete-event system simulation, 5th ed. Prentice-Hall, Upper Saddle River, NJ. <http://www.bcon.net>

# Artificial Intelligence Applied to the Geography: A Connectionist Approach



Mauro Preda

**Abstract** One of the critical points concerning the application of quantitative methods to geography consists in treating the map representation as the main object of analysis, rather than the territory directly. In this way, geographical elements in a dynamic n-dimensional space are studied locked in a static medium, albeit a digital one, deformed and flattened in only two dimensions, as in ‘Flatland’ the famous novel by Edwin A. Abbott. The connectionist approach, on the other hand, makes it possible to deal directly with geographical space and therefore the system-territory in its dynamicity and complexity (A cura di Gianluca Bocchi e Mauro Ceruti, La sfida della complessità, Feltrinelli, Milano, IT (1992)) in n dimensions, preserving the reciprocal function and relationship of the elements that compose it. Here, we will show how in the abstract the territory can be treated as a neural network, and thus lay the foundations for the application of artificial intelligence algorithms to geography, aimed at the development of a robust decision support system.

**Keywords** Connectionism · Quantitative geography · Artificial intelligence · Artificial neural networks · General systems theory · Decision support system

## 1 Introduction

For the sake of clarity, let us say at the outset that artificial intelligence (AI) [8, 10] and connectionism are both fields of study within computer science and cognitive science that deal with how machines are able to learn and solve problems. AI focusses on computing algorithms and building systems that are able to reason and make autonomous decisions. The main goal of AI is to create machines that can act as if they had human intelligence, solving problems of different kinds. Connectionism, on the other hand, focusses on the construction of artificial neural networks (ANNs), [8, 9] i.e. systems that simulate the functioning of the brains of living beings. Alan

---

M. Preda (✉)

Adjunct Professor (2000–2015), Università Cattolica of Milan, Milan, Italy  
e-mail: [mauro.preda@gmail.com](mailto:mauro.preda@gmail.com)

M. Turing's 1948 essay entitled 'Intelligent Machines' [10] surprisingly anticipates what would be years later the first and most interesting fields of application of AI, namely:

- (a) Various games, such as chess, filet, bridge, poker
- (b) Language learning
- (c) The translation of languages
- (d) Cryptography
- (e) Mathematics.

What do these very different fields of application have in common? We could quickly answer as follows: they require an intellectual effort to explore in depth all possible consequences of an action, as a decision support system; they also require concentration, memory, learning and continuous training, they are based above all on well-defined laws and rules while allowing the same problem to be tackled and solved in several and all valid ways.

Now, can geography join this list of fields of application? The answer is yes, and the necessary technical and theoretical foundations will be explained below: connectionism and systemic analysis, which we will discover have a link to a past that goes back to ancient Greek mythology.

In recent years, in both the academic and purely technological-commercial spheres, artificial neural networks have been enjoying increasing success in the most diverse fields of application, precisely because of their peculiar ability to solve complex problems. In fact, the basic idea behind ANNs dates back to the 1940s, when, in an attempt to build an intelligent system, as part of the research conducted on artificial intelligence, they developed architectures that were properly hardware, taking the natural neuronal structures of the brain of evolved beings as a model. In this regard, we must not forget the fundamental research conducted in the field of genetic psychology by Piaget, [1] which surprisingly effectively explicates the natural mechanisms of the learning process taken up and adapted in the computational systems proper to ANNs. In fact, Piaget's lesson argues that the mind, as a living organism, is endowed with self-regulating systems that tend to bring the structure back into balance, once an imbalance has occurred in one of its parts or the whole. Balancing thus becomes the fundamental factor in mental development because it does not simply tend to repair the perturbations caused by experience, but to re-establish equilibrium, taking into account both the perturbing factors and the mechanisms activated by them. Around the 1970s, therefore, a certain disinterest arose around research on ANNs, born from the conviction that they were unsuitable for solving particular computational problems and in particular classification problems. Recently, interest in ANNs has been growing stronger and stronger, the reasons for this being of various origins, which can be traced back to the advent of faster computers on which increasingly complex networks can be simulated, the interest in building massively parallel computers, and, in particular, the discovery of new architectures for neural networks and effective learning algorithms [9]. These

new architectures for neural networks have been termed ‘connectionist’ and are characterised by:

- A large number of simple, autonomous processing elements. (= neurons).
- A large number of connections (= synapses) weighted between elements: the weights encode the distributed knowledge of the network.
- Highly parallel distributed control.
- A trend towards automatic learning of internal representations. ANN models can be distinguished by architecture (number of units, number of levels, number and types of connections) and learning mechanisms. Based on the number of functional levels of the neural units, network models can be divided into two groups: (a) two-level networks, where the input units are directly connected to the output units, resulting in a direct relationship between the input and output signal forms; (b) networks with three or more levels, where one or more hidden levels are inserted between the input and output levels, useful in cases where the output forms are not so directly adaptable to the input signals (= nonlinear systems) and interconnections (totally connected, feedforward, feedback). The mode and direction of signal flow in the network allow networks to be distinguished into the following categories:
  - A fully interconnected network is when an output signal of a unit neural unit affects all the units of the higher level, becoming in turn input.
  - A feedforward network is when a layer only transmits its output signal to its next layer, the stability of the network is guaranteed in one step.
  - A feedback network allows each unit to receive an input signal, back from the next level, the adaptation process repeats itself until the desired level of stability is reached. The stable state, i.e. when an association between input and output is realised, may necessitate numerous oscillations and signal exchanges, which on the one hand increases computation time, but on the other allows complex associations to be established, which makes networks numerical systems capable of approximating any function.

## 2 The Territory as a Network: Its Origins in Myth

I will begin with an event that occurred outside of time and before space described by Ferecides of Syrus (sixth century B.C.) [5]: the sacred wedding between Heaven (Zas or Jupiter) and Earth (Chthon from which chthonic means underground, invisible, etc.). The bride Cton appears veiled, she removes her veil and Heaven immediately covers her with a cloak that he himself has embroidered for her, now the bride is called Gé from which precisely, Gaia she who shines, shines. The weave and design of this cloak are very intricate, it is finely embroidered and is, moreover, richly coloured, the same colours that we can easily admire on every Atlas.

Ctòn is now Gé, conceptually the two terms are in opposition but in reality there is an intimate relationship between them that only through geography can we understand. We are confronted with two distinct but closely related realities: one guides us comfortably on the surface and allows us to see and measure Gaia, which shines and shines in the sunlight, the other takes us, on the other hand, into the dark and subterranean, chthonic, and therefore invisible, zones. These dynamic, continuous, borderless and elusive spaces are, e.g. those of culture, religion, language, thought and therefore politics, security, etc. In this continuous dialectical relationship between visible and invisible, material and immaterial, the territory itself is realised through information networks. Of course, to be a little more technical, traditional metrics and the albeit sophisticated GIS information tools we use, if not adequately redesigned, risk being insufficient to capture the multiform and multidimensional reality of the territory. We are in a context where the new global economy has in fact shortened distances almost cancelling them.

The cloak that Heaven makes for the bride Gaia is made of a richly decorated and embroidered fabric, throughout the process it is reasonable to imagine that it is kept tightly stretched on a loom. The basic notions of weaving lead back to a principle: how a simple thread can be transformed into a fabric, that is, translated into geometric-topological terms, how to go from dimension ONE to dimension TWO or rather from LINE to PLAN. The transformation takes place thanks to the weave that forms a dense and tight network. Going further, the cloak that has just been removed from the frame to cover Gaia's body, thus acquires shape and volume, acquires an extra dimension by passing to dimension THREE, i.e. space.

Metaphorically, if we look closely at the fabric of the mantle, we discover that it is held together and is functional precisely because of its network structure, which is in fact transferred to Gaia. At this point, in the midst of the 'age of networks' and in the light of the extraordinary expansion of global communication networks and the thickening of individual connection nodes, let us try to reflect on how many and what concrete consequences the 'network phenomenon' can have on the territory.

Now, of this new and extraordinary tension towards global communication, a phenomenon that affects society, religion, art, language, culture, economics, politics, national security, etc., it is up to geography, already defined as the 'science of relations in space', to provide us with the possible conceptual tools suitable for its analysis and interpretation, so that it can be used positively, more productively and possibly more intelligently. Indeed, precisely in this period of globalisation, geography could rediscover its ancient origins and identity by returning to deal precisely with what by its nature is invisible or unrepresentable, and together with it, cartography would thus return to being a more conceptual tool, a model for a reference that is no longer Euclidean, static with absolute coordinates but not Euclidean, dynamic, relative and relational.

### 3 From System-Territory to Network-Territory

Assuming as the object of geography, the territory understood as an organised spatial reality, the territory is configured as a real space that is defined through the aggregate of objects that compose it. The objects with which the territory is configured and defined are in a reciprocal relationship [6] insofar as they coexist for the attainment of an end, in fact they themselves are defined according to their respective type, entity, size, distribution, geolocation, all criteria with which it is possible to establish and interpret the relationships [4] that reciprocally connect them: the territory can thus be treated as a system. The level of complexity [2] of such a system increases in relation to the numerosity of the observations and therefore of the data introduced (entification) between them, where the type, direction and weight (quantification) of the variability over time define and qualify the relations, configuration and stability of the territory system. At this point, to better represent logically and cartographically the individual components of the system, as well as the system itself, a change of cartographic paradigm is required: from metric, absolute, cartesian to relative. In this sense, here is what Leibniz wrote to Clarke in a letter dated 25 February 1716, taken from the Leibniz-Clarke Epistolary: For I have noticed more than once that I consider space as something purely relative, not otherwise than time; as an order of consistencies, in the same way that time is the order of successions. For space marks, in terms of possibility, the order of those things existing at the same time, insofar as they exist together, without entering into their particular modes of existence. And as one sees several things together, one grasps that order of things among them [7]. This way of representing geographical objects is in fact the most natural; spatial perception is, in fact, the ability of evolved living beings to be aware of their relationship to their surroundings and their relationship to themselves. Spatial perception is made up of two processes: the first are those that construct representations of the space around us through the senses, the second are the representations with respect to our body, such as position or orientation. Space is first of all what surrounds us: objects, elements, people, etc. Space is also part of our way of thinking, since it is where we gather all the data from our lived experience. Spatial perception allows us to define the layout of our environment and our relationship to it, to understand the relationship of objects when there is a change of position in space. It helps us to think in multiple dimensions (distance time, space, cost, meaning, etc.), and allows us to visualise objects from different angles that we recognise regardless of the perspective from which we view them. To give a practical example, if I pick up a pen from my desk and then put back to the place where it was before, the ‘where’ is defined not by absolute coordinates (longitude and latitude) but by relative, semantic, logically deductive references: example in the pen glass! Relative spatial perception helps us to move and orient ourselves in places we have not yet explored but which are somehow part of our experience of similar places we already know. Let’s take another simple practical example: railway stations have a very similar structure, some are just an exact copy of others, they are in different places but this does not create problems in orienting oneself within them, even for those arriving there for the first time!

Of course, a relative geolocalisation system could be implemented in GISs alongside the traditional geolocalisation model. This would make it easy to solve problems related to the distribution model, the search for spatial elements, structures, and events that, although related to different locations, are more similar. Modern graph databases could be the most suitable component for the development of a relative co-ordinate GIS.

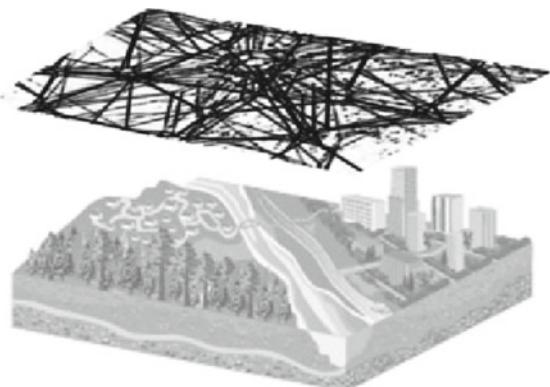
#### 4 The Invisible ‘Neural Network’ of the Territory

Taking the aforementioned systemic theory applied to the territory to its extreme consequences, together with the advantages and concrete applicative possibilities of a reticular spatial data processing model, in order to adapt AI to geography, it is necessary that the object of study of geography lends itself to being defined by a suitable data and algorithm model so that descriptive, predictive solutions can be arrived at from a highly complex system-territory. The ANN and its territory is represented in Fig. 1.

Artificial neural networks are now able to simulate the functioning of a living being’s brain. What if, in the abstract, the territory itself were to be treated as a living being and thus endowed with a learning and intelligence capacity? The new concept of ‘territorial intelligence’ [3] as an innovative network organisation of information and knowledge useful for the development and competitiveness of a territory fits into this direction. It is a socio-territorial component, which acts on the basis of the knowledge and human resources available in the territory, in order to achieve goals and objectives shared by the community. It deals with the valorization of resources, extending awareness of them through forms of communication that involve the various actors active in the territory: environment and society.

Territorial intelligence is the only one capable of making available resources known and promoting communication that brings citizens closer to common needs.

**Fig. 1** ANNs and territory



In this sense, governance comes into play, as different actors enter the scene and a transfer of competences develops between local actors of different orientations. In practice, there is a territorial intelligence capable of safeguarding community interests. A new way of conceiving the public good is born, it is therefore a cultural issue, as is the conflict of powers and the lack of communication between the grassroots and the political summit. Active citizenship is thus activated, which conceives of the conservation of the territory as equivalent to the conservation of one's own well-being; a co-participation, a cooperation, a sharing of problems and needs is thus developed. Territorial intelligence wants to find a way to respond to these needs without exploiting or drying up territorial resources, enhancing the specificities of its own territory, to define models that do not claim to be one-size-fits-all; in short, it is concerned with sustainable development.

The concept of territorial intelligence has had a recognised presence in the geographical, sociological and political debate for some years now. Now we need to verify whether the territory can be configured in the abstract as an intelligent tissue such that it can be treated as an artificial neural network with all its properties.

Of course, this is a puzzling hypothesis and proving such a theory is very difficult. To eliminate initial perplexity, I used a simple stratagem: I posed a few questions to ChatGPT, a well-known impartial and sufficiently trained AI 'tool'.

#### ***4.1 Can the Territory Be Analysed with a Connectionist Approach?***

Yes, a territory can be analysed with a connectionist approach, which is a type of computational modelling that is often used to study complex systems that exhibit emergent properties and behaviours.

The connectionist approach is based on the idea that complex systems are made up of interconnected components that interact with each other to generate emergent properties and behaviours. This approach is often used in the study of artificial neural networks, which are computational models that are inspired by the structure and function of biological neural networks.

In the context of a territory, the connectionist approach can be used to model the interactions between different components of the system, such as natural resources, infrastructure, and human populations. This can involve constructing computational models that simulate the behaviour of the different components of the system, as well as the interactions between them.

The connectionist approach can be useful for understanding the complex behaviours that emerge from the interactions between the different components of a territory. For example, a connectionist model could be used to simulate the effects of changes in the climate or natural resource availability on the settlement patterns and economic activities of human populations.

However, it is important to note that the connectionist approach is a computational modelling technique, and the models that are constructed using this approach are simplifications of the complex real-world systems that they are intended to represent. As a result, the results of connectionist models should be interpreted with caution, and should be validated and tested against real-world data whenever possible.

## ***4.2 Does the Territory in the Abstract Have an Intrinsic Memory and Intelligence?***

In the abstract sense, a territory does not have an intrinsic memory or intelligence like a living organism or an artificial neural network. The concept of memory and intelligence is closely linked to cognitive processes, and it is not appropriate to attribute these characteristics to a non-living system like a territory.

However, a territory can exhibit certain types of memory and intelligence-like behaviours in the sense that it can adapt and learn from its past experiences. For example, a territory can ‘remember’ the effects of natural disasters or human activities that have occurred in the past, and this memory can influence the way in which the territory evolves and develops in the future.

Similarly, a territory can exhibit a form of ‘intelligence’ in the sense that it can respond to changing environmental conditions and adapt to new circumstances. For example, a territory can respond to changes in the climate or natural resource availability by shifting its economic activities or altering land use patterns.

Overall, it is important to be cautious when using terms like memory and intelligence to describe the behaviours of non-living systems like territories, as these terms are more commonly associated with living organisms and artificial intelligence systems. However, the analogy between a territory and a living organism or an artificial neural network can be useful for understanding the complex behaviours that emerge from the interactions between the different components of the system.

## ***4.3 Can the Territory as a System Learn?***

In the abstract sense, a territory can exhibit learning-like behaviours, although it is important to note that this is a metaphorical use of the term ‘learning’ and does not imply that the territory has a cognitive process similar to that of a living organism or a machine learning algorithm.

In the context of a territory, ‘learning’ can refer to the ability of the system to adapt and respond to changing conditions over time. For example, a territory can learn from the effects of past natural disasters or human activities, and this learning can inform future decision-making and planning. Can the territory in the abstract be seen as a neural network?

In the abstract sense, it is possible to view the territory as a neural network. Both a territory and a neural network are complex systems made up of interconnected components that interact with each other to generate emergent behaviours.

In a neural network, individual neurons are connected to each other through synapses, and their interactions generate complex patterns of activation that give rise to the network's behaviour. Similarly, a territory can be seen as a complex system of interconnected components such as physical features, natural resources, and human-made structures, which interact with each other to generate emergent properties and behaviours.

Furthermore, the concept of feedback loops and self-organisation in neural networks can also be observed in territories. For example, the interactions between the physical features of a territory, such as water sources and topography, can influence the distribution of natural resources, which in turn can affect human settlement patterns and infrastructure development.

However, it is important to note that the analogy between a territory and a neural network is only a metaphorical one and does not imply that the two systems operate in exactly the same way. While both systems can exhibit emergent properties, a territory is a much more complex and dynamic system than a neural network, and it is influenced by a wide range of factors that are not present in the case of a neural network.

Additionally, the interactions between different components of a territory, such as natural resources, infrastructure, and human populations, can result in emergent behaviours and patterns that can be seen as a form of 'learning' at the system level. For example, the growth of a city over time can be seen as a result of the accumulation of knowledge and experience that is passed down from one generation to the next.

However, it is important to note that these learning-like behaviours are the result of the interactions between the components of the system, rather than an intrinsic property of the territory itself. The ability of a territory to adapt and learn is ultimately dependent on the actions and decisions of the individuals and organisations that inhabit and use the territory.

#### **4.4 Does the Territory Have Memory?**

In the abstract sense, a territory can exhibit memory-like behaviours, although it is important to note that this is a metaphorical use of the term 'memory' and does not imply that the territory has a cognitive process similar to that of a living organism or a computer.

The concept of memory in the context of a territory refers to the ability of the system to retain information about past events and experiences. This can include physical features such as geological formations, water sources, and natural resources, as well as human-made structures such as buildings, roads and other infrastructure.

This 'memory' of past events and experiences can influence the way in which a territory evolves and develops over time, e.g. the location of water sources and

other natural resources can influence the settlement patterns of human populations, which in turn can affect the development of infrastructure and other human-made structures.

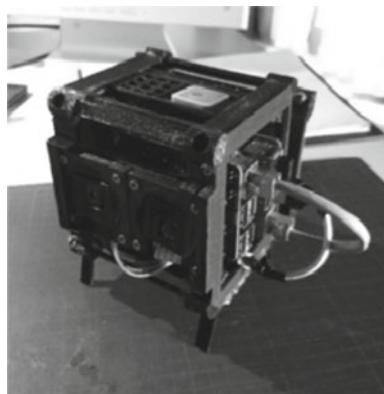
Additionally, past events such as natural disasters, conflicts, and economic downturns can have a lasting impact on a territory and its inhabitants, and can shape the way in which future events and decisions are perceived and acted upon.

Overall, the concept of memory in the context of a territory is a metaphorical one that refers to the ability of the system to retain information about past events and experiences. This ‘memory’ can influence the way in which the territory evolves and develops over time, but it is ultimately dependent on the actions and decisions of the individuals and organisations that inhabit and use the territory.

## 5 Discussion and Conclusion

ChatGPT suggests that the connectionist model hypothesis applied directly to the territory for the use of AI is well-founded. Treating the territory in the abstract as a neural network thus enables the application of ‘pattern recognition’, i.e. automatic training and learning models, which draw the necessary data from current ‘big data’ for the recognition of e.g. repeated and distributed spatial patterns, for simulations, predictions and thus for an efficient neural and spatial decision support system (NSDSS). The next step will be the development of a new version of the NSDSS prototype, which will be trained on a restricted spatial area used as a sample, a very first example, although very rudimentary, I presented at the FOSS4G ITALIA 2019 conference, with a paper entitled ‘Geodeeplearning Machine: Presentation of the Prototype’. The aim of the project was the realisation of a device capable of recognising geographical objects in the work environment, georeferencing them and relating them to each other according to spatial and semantic rules, in order to create a network of n-dimensional relations. The tool realised and called ‘geo deep learning machine’ consisted of an assembly of low-cost hardware components and software in the PYTHON language developed by the author. The very small device was capable of operating in the visible and thermal infrared, as it had two cameras: a common RGB webcam and a micro-camera. The calculation unit consisted of two raspberry PIs operating in parallel, see Fig. 2.

**Fig. 2** Geo deep learning machine: prototipo (2019)



## References

1. Axia Giovanna (1986) *La mente ecologica*, Giunti Barbera, Firenze, IT
2. A cura di Gianluca Bocchi e Mauro Ceruti, *La sfida della complessità*. Feltrinelli, Milano, IT (1992)
3. Bertacchini Y, Paul Déprez P, Paul Rasse P (2014) The territorial intelligence process: ecology of communication for development of hybrid territories. *Comput Inf Sci* 7(4)
4. Emanuela C (2001) *L'ordine del mondo e la sua rappresentazione*. UNICOPLI, Milano, IT
5. Franco F (2009) *La crisi della ragione geografica*. Einaudi, Torino, IT
6. Giorgio I (1996) *La visione matematica della realtà*. Bari, IT, Laterza
7. Leibniz GW (1967) *Scritti filosofici*, vol I, trad. di D. Omero Bianca, Torino, IT, UTET
8. Openshaw S, Openshaw C (1997) *Artificial intelligence in geography*. Wiley, Chichester, UK
9. Parisi D (1989) *Intervista sulle reti neurali*. Il Mulino, Bologna, IT
10. Turing AM (1994) *L'intelligenza meccanica*. Bollati Boringhieri, Torino, IT

# Artificial Intelligence, Administrative Proceeding, Protection and Enhancement of Cultural Property



Livio Perra

**Abstract** In this chapter, the author analyzes the possible applications of artificial intelligence in the protection and enhancement of cultural property. For this purpose, first it is necessary to identify a definition of artificial intelligence, then to examine the relationship between administrative activity and artificial intelligence. Later, attention is focussed on cultural property and the possible application of artificial intelligence in this sector. The analysis carried out by the author leads to the importance of the use of artificial intelligence also in the sector of protection and enhancement of cultural property. Artificial intelligence can help in activities concerning the protection and enhancement of cultural property, albeit with the consideration that it cannot replace human beings, but can perform some tasks under human supervision and control. In this sense, artificial intelligence can become a tool/collaborator of human beings in these activities.

**Keywords** Artificial intelligence · Administrative proceeding · Cultural property · Protection · Enhancement · Tool/collaborator

## 1 Introduction

This chapter is dedicated to the analysis of the possible applications of artificial intelligence in the administrative proceeding and the sector of protection and enhancement of cultural property. Specifically, the advantages, disadvantages and possible limitations of application in these sectors are analyzed. Artificial intelligence can concretely constitute a tool that helps public administrations in carrying out their tasks and becomes an aid in the protection and enhancement of cultural property. The attention is focussed on the range of possibilities that can be opened up by the use of ever new forms of artificial intelligence. Thanks to technological progress,

---

L. Perra (✉)

Dipartimento di Storia, Scienze dell’Uomo e della Formazione, Università degli Studi di Sassari, Sassari, Italy

e-mail: [lperra@uniss.it](mailto:lperra@uniss.it)

human beings have always found in new tools the necessary aid to carry out their tasks. Artificial intelligence can, in this sense, help in the management of large amounts of data, in the organization of the data itself, in assuming useful information in administrative proceedings. Specifically, it is a tool that collaborates with the public functionary (natural person) and not a tool that replaces functionaries in the entire activity carried out by the human being. A total substitution in the process of decision-making of the public administration could be in contrast with the principles that regulate the administrative proceeding, the obligations of public administrations and the guarantees envisaged for each decision that impacts on subjective legal situations, legitimate interests and rights of citizens. Therefore, entrusting the outcome of an administrative proceeding to algorithmic decisionism would not seem to be a practicable way. Human intervention should always be present. It should take the form of human supervision and continuous interfacing between human beings and machines.

In the cultural property sector, the verification of cultural interest and the declaration of cultural interest, even if aimed at the general interest of safeguarding and protecting cultural property, impact on the situations of third parties. With specific reference to the declaration of cultural interest, it must be highlighted that the evaluation of the presence of cultural interest involves limits and constraints for private individuals. Similarly, in these sectors it would seem difficult to replace a public functionary (natural person) with an artificial intelligence that performs the same tasks, reaching an algorithmic decision-making process. It would be difficult for the public administration to motivate its decisions, explaining all the steps that led to their adoption.

The activities of protection and enhancement of cultural property can, however, be integrated by tools such as artificial intelligence. It will be able to contribute to the management of data and information, simplifying the tasks of human beings. Furthermore, thanks to artificial intelligence it will be possible to control more quickly how the protection of cultural heritage takes place, to group all similar properties to facilitate the activity of undertaking common protection actions. The valorization of cultural heritage is subject of concurrent legislative power of the State with the regions, pursuant to Title V of the Italian Constitution (see Article 117 of the Italian Constitution). The artificial intelligence could be a useful option in the screening and coordination of common interregional initiatives. With regard to the enhancement of cultural properties, understood as the knowledge and fruition of the properties, it can be observed that IT tools are currently able to contribute to greater knowledge and fruition remotely. From this point of view, artificial intelligence could be conceived and used to increase these new forms of valorization of cultural property and interact, through a series of questions and answers between users and artificial intelligence, to make knowable cultural properties, whose users are unaware of its existence.

## 2 Artificial Intelligence

The term artificial intelligence was coined by John McCarthy in 1956, during a conference organized at Dartmouth, USA [1].

Over the years, the topic has captured the interest of many scholars and various definitions of artificial intelligence have been formulated. For the purposes of this chapter, it is necessary to identify a definition of artificial intelligence, which includes various aspects. For this reason, the author chooses to refer to the definition presented by Lasse Petteri Rouhiainen [2]. In his seminars, he defines artificial intelligence as “the ability of computers to perform activities that normally require human intelligence” [2].

It should be noted that artificial intelligence should not be understood as a mere copy of the human brain and its functioning. The tasks entrusted to artificial intelligence can be completed without an emulation of human intelligence. Artificial intelligence uses some methods to process information and to solve more or less complex problems [3]. Furthermore, it must be borne in mind that there are different types of artificial intelligence that can combine multiple methods based on the activity they have to perform [3].

In this regard, it is useful to recall what Lasse Petteri Rouhiainen [2] affirms. He underlines that artificial intelligence finds various applications and the range of situations in which it can be used is currently growing [2]. Among them, he identifies “still image recognition, classification and labelling” [2], “the improvement of the implementation of algorithmic trading strategies” [2], “efficient and scalable processing of patient data” [2], “predictive maintenance” [2], “object detection and classification” [2], “the distribution of content on social networks” [2] and “protection against cybersecurity threats” [2].

## 3 Administrative Activity and Artificial Intelligence

New technologies have often played an important role in human life. They have helped to speed up some activities. They have helped human beings in carrying out some tasks by reducing the expenditure of energy and provided tools that facilitate the management of big units of data. It is now necessary to understand how the activity of the public administration can benefit from the use of artificial intelligence, what are the limits, the advantages and disadvantages.

In order to use of artificial intelligence in public administration activity, it is useful to observe, as Diana-Urania Galetta and Juan Gustavo Corvalán [3] point out, that there are activities that can be standardized, which can be automated, and activities where automation is not possible.

Specifically, in administrative proceedings artificial intelligence can contribute to the collection, arrangement, connection of data and information and their management. In this way, it becomes a means, an instrument at the service of the public

administration, of the human being in carrying out the investigation phase and adopting the administrative measure [3].

A necessary distinction must be made with reference to the activity of public administration between restricted and discretionary activity. Following the reasoning of Diana-Urania Galetta and Juan Gustavo Corvalán [3], an automation in the restricted activity of the public administration would lead to a standardization of the procedural activity which would guarantee the impartiality of the public administration and there would be a reduction of time required for adoption of administrative measures.

Regarding the discretionary activity of the public administration, in agreement with what Diana-Urania Galetta and Juan Gustavo Corvalán [3] affirm, the algorithmic decision could raise questions with specific regard to respect of administrative principles, of the obligations of public administration and more generally undermine the very relationship between public administration and citizens and damage the nucleus of dialogue and participation in the administrative proceedings. In some cases, it is difficult to identify which elements have influenced and determined the algorithmic processes of decision-making. Public administration should succeed in the arduous task of explaining how a machine has defined the content of the administrative provision [3].

For these reasons, as considered by the Regional Administrative Tribunal for Lazio on several occasions [4–11] and underlined by Diana-Urania Galetta and Juan Gustavo Corvalán [3], it would not seem a feasible option to replace with an algorithm the official (natural person) in the investigation phase and in adopting the administrative measure. In this way, artificial intelligence cannot replace the human being, but provide help, become a tool, or support the administrative activity.

## 4 Cultural Property

The cultural heritage sector has its own specificity, where the protection and enhancement of the properties are fundamental.

The first step is to understand what the term cultural heritage includes. In the Code of Cultural Properties and Landscape, cultural properties and landscape properties constitute cultural heritage (see Article 2, paragraph 1 of the Code of Cultural Properties and Landscape). The Code of Cultural Properties and Landscape identifies cultural properties as things that have an artistic, historical, archaeological or ethno-anthropological interest. The presence of the interest is required in different form in the article 10 (simple presence in the first paragraph, properties presenting a cultural interest *ex se* in the second paragraph and a particularly important or exceptional interest in the third paragraph).

For immovable or movable things, which are the work of an artist who is no longer living, created for over seventy years, which belong to the state, to the regions, to other territorial public bodies, to other public bodies and institutes, to private legal entities which do not pursue profit-making and to civilly recognized ecclesiastical

bodies, indicated in the first paragraph of article 10 of the Code of Cultural Properties and Landscape, the verification of the cultural interest of the article 12 is envisaged. The competent bodies of the ministry with this procedure verify the presence of cultural interest.

Article 13 and following of the Code provide the declaration of cultural interest of immovable and movable things, indicated in the paragraph 3 of article 10 of the Code of Cultural Properties and Landscape. This procedure of declaration verifies the presence of interest in the intensity required by article 10, paragraph 3 of the Code of Cultural Properties and Landscape (particularly important or exceptional) [12]. The forecast of an intensity of cultural interest, expressed in terms of particular importance or exceptional interest, indicates that the evaluation of the existence of the interest is subject to a technical-discretionary assessment [12]. The objects, in which the presence of interest in the extent required by law is ascertained, are brought back into the category of cultural property and are subject to the related protection [12].

## 5 Artificial Intelligence in the Cultural Property Sector

It is now necessary to consider whether it is possible to use artificial intelligence in the cultural property sector. A total automation of administrative proceedings would seem difficult to configure in the cultural property sector, e.g. the declaration of cultural interest is an administrative proceeding. The outcome can take the form of a provision that imposes constraints or limitations on the private individuals to whom the cultural properties belong. These measures, which arrive following a technical-discretionary evaluation, can be reviewed exclusively with reference to their motivation. The judge will be able to verify the adequacy and logic of the motivation [12, 13]. Entrusting this procedure to an algorithmic decisionism could produce some difficulty to understand and explain on which bases and steps was reached a certain result, as in other administrative proceedings as affirmed by Diana-Urania Galetta and Juan Gustavo Corvalán [3]. Artificial intelligence can certainly contribute to help the subjects who have to make these decisions, making various cognitive elements available, also through different data connections, but human-machine interaction is always necessary in order not to depersonalize the decisions and also allow a traceability of the decision to the entity and to responsibility of natural persons [3].

Several opportunities for the use of artificial intelligence in the cultural property sector are opening up with specific reference to the help it can provide in the management and connection of large amounts of data.

Now it is necessary to analyze the definitions of protection and enhancement of cultural property.

The term protection refers to activities aimed at the recognition, conservation and protection of cultural properties [12] and the term enhancement includes all activities that lead to a better knowledge and conservation of cultural properties and increase their fruition [12].

The Italian Constitution, following the reform of title V, established the legislative power of the State for the protection of cultural heritage and the concurrent power of the State and the regions for their enhancement [12].

In the field of protection and enhancement, numerous uses of artificial intelligence are conceivable and they will increase over the years, thanks to the progress of technology. Artificial intelligence can become a tool and an aid in activities relating to protection and enhancement of cultural property. Certain procedures may be automated, albeit always under careful human supervision. Some systems could lead to the possibility of checking the state of conservation of cultural properties more and more quickly. Other systems could be used in the planning phases of actions aimed at the protection of cultural properties. Artificial intelligence could identify the properties that have similar characteristics on which to focus attention by grouping them into categories, and intervene with the same tools. Other systems could analyze and suggest different hypotheses about the protective actions and predict a possible effectiveness of these, in order to help humans in the choice between different feasible actions.

In the enhancement sector, artificial intelligence could connect different cultural properties of different regions to undertake common initiatives. Other systems could contribute to making cultural properties more accessible through remote visits. Other systems could allow an enhancement of the tools with which to observe objects remotely without damaging the originals. If current computer systems can already make many sites and artefacts known thanks to remote mode even to physically distant people, some systems could interact with users and show them other unknown sites and artefacts that have similar characteristics. Artificial intelligence, as is already the case with simpler IT tools, could find numerous developments in bringing knowledge of cultural properties to a wider audience without spatial boundaries.

An important example of the contribution that can be offered by artificial intelligence in the cultural property sector can be observed in the use of Watson technology in the Pinacoteca de São Paulo, Brazil. This experience, with specific reference to the use of artificial intelligence and educational processes in museums, is described and analyzed by Milene Chiovatto [14].

In 2016, the International Business Machines (IBM) proposed the use of the Watson system to the Pinacoteca of São Paulo [14]. In particular, it is a system that “uses artificial intelligence (AI) in order to simulate dialogues with visitors” [14].

Milene Chiovatto [14] describes all the phases of the project. Initially, seven works of art were chosen: Saudade, realized in 1899 by José Ferraz de Almeida Junior; O Porco, realized in 1967 by Nelson Leirner; Lindonéia, A Gioconda do Subúrbio, realized between 1966 and 1968 by Rubens Gerchman; Bananal, realized in 1927 by Lasar Segall; Ventania, realized in 1888 by Antonio Parreiras; São Paulo, realized in 1924 by Tarsila do Amaral; O mestiço, realized in 1934 by Cândido Portinari. After the phase of selecting the works of art, the second phase consisted in the development of a digital platform for the collection of questions (including names of the authors of the works, possible interpretations of the works), creating possible answers from various data, (e.g. books, educational materials) and organizing the various questions according to their similarity. Furthermore, when the answer to the visitor’s question

does not match those of the repertoire, the answer is "I don't know". Questions are asked through an interface or microphones of smartphones made available to the public [14].

Now, after having understood the functioning of the Watson technology, it is necessary to observe what considerations have emerged from the use of this system in the Pinacoteca of São Paulo.

Among the considerations about the use of Watson technology in the Pinacoteca of São Paulo [14], it emerges the possibility offered to visitors to take advantage of new and additional tools that favour contact and learning in museums through a new form of interaction between human being and machine. In particular, a real dialogue on cultural property is established between people and machine. The use of new technology attracts new visitors who do not normally access the museums. At the same time, although the use of Watson technology does not capture the interest of children under the age of seven, as they prefer traditional learning activities (games and other activities) and it does not arouse the interest of elderly visitors, this new method is appreciated by most visitors. Numerically, it is found that in the project period (about one year and three months), of the 80,000 visitors, 25,000 used Watson and only 15% of the answers provided by the Watson system were not considered satisfactory [14].

Now, starting from the description of the Watson project and considerations about this experience by Milene Chiovatto [14], numerous food for thought can be drawn.

The Watson system opens up various possibilities for interaction between human being and machine and becomes a tool capable of promoting learning in museums.

The use of new technology intrigues some visitors who are not regulars at museums. This fact brings and promotes knowledge of cultural property to a wider audience. Watson is educated by predicting possible questions and their answers, and the results are evaluated by humans.

The potential of systems such as Watson can be placed at the service of museums, as they help welcome the public and accompanied people on their visit. The visit is personalized, to the extent that each visitor using the functions of the Watson tool can receive answers to questions in real time and contemplate the work of art together with the baggage of knowledge underlying it.

The experience took place with reference to seven works, but the potential observed suggests a possible future use on a large scale. The contribution of the human being is not lacking in the Watson experience. Training the Watson system with the necessary possible questions and answers took time and energy. Furthermore, as stated by Milene Chiovatto [14], these technologies are currently expensive. For this reason, their use on a large scale would be more predictable when technological progress makes these technologies more economically accessible.

When asked whether these technologies can replace the role or activity of human beings, a negative answer seems to have to be given, as there is a slice of the public (children under the age of 7 and elderly) who prefer traditional activities. It should also be noted that 15% of the answers were found to be unsatisfactory and therefore in those cases the intervention of human beings is necessary. Furthermore, the intervention of human beings is still essential today to supervise the operation of

systems such as Watson and allow for constant improvement. In particular, we can observe the contribution that these systems provide, facilitating the fulfilment of human activities.

Artificial intelligence opens the door to wide possibilities. It could support activities related to protection and enhancement of cultural properties. It should not supplant the role of humans in these activities, but it could become an important tool/collaborator.

## 6 Conclusion

New technologies enter people's lives by helping to lighten daily tasks and carry them out more quickly. In this chapter, the author analyzes the possible applications of artificial intelligence in administrative proceedings and in the cultural property sector, with particular attention to the protection and enhancement of cultural properties.

A total automation of administrative proceedings would seem difficult to configure, as replacing the discretionary activity of the public administration with an algorithmic decisionism. This could raise questions relating to the respect of the principles that govern and direct the administrative proceeding and the obligations of public administrations. Similarly, it is noted that full automation is not possible in the verification of cultural interest and in the declaration of cultural interest. The declaration of cultural interest is an administrative proceeding whose outcome, also known as a restriction provision, involves restrictions and limitations for private subjects to whom the cultural properties belong. This declaration comes following the completion of the administrative proceeding, in which a technical-discretionary evaluation takes place. Replacing the official (natural person) would raise the same issues related to other administrative proceedings, such as the difficulty of motivating algorithmic decision-making.

Artificial intelligence could provide an aid in the protection and enhancement of cultural property. Specifically, it could help human beings in the management of large volumes of data, in correlating them and, more generally, contributing to the cognitive phase to be used as a basis for decisions. Furthermore, in the enhancement of cultural property, it could both facilitate the formation of joint actions between regions and make a strong contribution to the dissemination of knowledge of cultural properties and remote fruition. If there are already IT tools with which geographically distant people can learn about cultural property and use virtual tools to visit the sites, this could reach a further level thanks to artificial intelligence, e.g. artificial intelligence could be conceived to relate different cultural properties, also to favour the knowledge of cultural properties and sites of which the user is unaware even of the existence. Artificial intelligence could contribute to enhancement, understood as spreading knowledge of cultural properties and allowing it to be used remotely. It would become a tool that would lead users remotely to new places to visit, to discover unknown sites with characteristics similar to those visited.

Therefore, the application of artificial intelligence could facilitate some activities that are part of the administrative proceeding and in the sector of protection and enhancement of cultural property.

If for the restricted activity there would be no problems in an automation (always under the supervision of human beings) that leads to a speeding up the execution of tasks, it is more complex to think of automation in the public administration's discretionary activity. In such cases, artificial intelligence could make its contribution by helping humans perform some tasks. It would not be a total replacement of the human being. Artificial intelligence could find expression in a collaboration/help of which the human being has the supervision and control.

## References

1. Reali Costa AH, Nunes de Barros L, Oliveira Rezende S, Simão Sichman J, Neri H (2021) Trajetória acadêmica da Inteligência Artificial no Brasil. In: Cozman FG, Plonski GA, Neri H (eds) Inteligência artificial: avanços e tendências. Instituto de Estudos Avançados da Universidade de São Paulo, São Paulo, pp 30–66. Available online <https://www.livrosabertos.sibi.usp.br/portaldelivrosUSP/catalog/book/650>. Accessed 31 Mar 2023
2. Rouhiainen LP (2018) Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro. Alienta Editorial, Barcelona
3. Galetta D-U, Corvalán JG (2019) Intelligenza Artificiale per una Pubblica Amministrazione 4.0? Potenzialità, rischi e sfide della rivoluzione tecnologica in atto. Federalismi.it 3:1–23. Available online <https://www.federalismi.it/nv14/articolo-documento.cfm?Artid=38014>. Accessed 31/03/2023
4. TAR Lazio, 10 Settembre 2018, n. 9224. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
5. TAR Lazio, 10 Settembre 2018, n. 9225. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
6. TAR Lazio, 10 Settembre 2018, n. 9226. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
7. TAR Lazio, 10 Settembre 2018, n. 9227. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
8. TAR Lazio, 10 Settembre 2018, n. 9228. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
9. TAR Lazio, 10 Settembre 2018, n. 9229. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
10. TAR Lazio, 10 Settembre 2018, n. 9230. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
11. TAR Lazio, 9 Novembre 2018, n. 10828. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
12. Ferretti A (2019) Manuale di diritto dei beni culturali e del paesaggio. Edizioni Giuridiche Simone, Napoli
13. Consiglio di Stato, sez. VI, 24 marzo 2003, n. 1496. Available online <https://www.giustizia-amministrativa.it>. Accessed 31 Mar 2023
14. Chiovatto M (2019) Watson, uso de Inteligência Artificial (AI) e processos educativos em museus. Revista Docênciia Cibercultura 3(2):217–230. Available online <https://www.e-publicacoes.uerj.br/index.php/re-doc/article/view/40293>. Accessed 23 Apr 2023

# Personalized Surgical Planning in Liver Surgery Using Virtual 3D-Models



Teresa Perra and Alberto Porcu

**Abstract** Liver surgery requires personalized preoperative planning, especially in case of major liver resections. In this chapter, we explore how AI-based technologies can help hepatopancreatobiliary surgeons to plan the best surgical procedure for each patient. In particular, we analyze the role of virtual 3D-models in surgical planning, describing the current practical applications, future perspectives, and challenges in this field. Virtual 3D-models allow us to establish the correct surgical indication for each patient, plan the amount of liver to be resected, and perform a tailored surgery.

**Keywords** Artificial intelligence · General surgery · Surgical planning · Liver surgery · Hepatopancreatobiliary surgery · Virtual 3D-models

## 1 Introduction

Artificial intelligence (AI) is becoming part of our daily lives in many different ways. In the last decades, AI has also begun to be incorporated into medicine to improve surgical care. Despite the technological advancements in image processing, two-dimensional (2D) images are not always sufficient to define the anatomical complexities of human body.

In this chapter, we explore how AI-based technologies, as virtual three-dimensional (3D)-models, can help hepatopancreatobiliary (HPB) surgeons to plan the best surgical procedure for each patient. Virtual 3D-models are very useful to visualize the hepatic lesions and their relationships with the surrounding structures in the preoperative surgical planning. They also help surgeons to plan the amount of liver to be resected, and perform a tailored surgery.

In particular, in Sect. 2, we focus our attention on virtual 3D-models and their preliminaries. In Sect. 3, we provide an overview about the personalized surgical planning in major liver resection. In Sect. 4, we analyze the role of AI in liver surgery

---

T. Perra (✉) · A. Porcu  
Azienda Ospedaliero Universitaria di Sassari, Sassari, Italy  
e-mail: [teresaperra92@tiscali.it](mailto:teresaperra92@tiscali.it)

and the practical application of virtual 3D-models in surgical planning, describing some practical examples. In Sect. 5, we describe the possible future perspectives and challenges in this field, focusing on the major limitations and advantages of using 3D-models in planning liver surgery. Finally, in Sect. 6, we expose our conclusions.

## 2 Virtual 3D-Models and Their Preliminaries

In the preoperative surgical planning, most HPB surgeons use two-dimensional (2D) images from computed tomography (CT), and magnetic resonance imaging (MRI) scans. These are useful to evaluate the hepatic lesions and their relationships to the surrounding structures, although two-dimensional (2D) images are not always sufficient in surgical planning. Sometimes surgeons need to better define the complexities of their patient's anatomy. Mental reconstruction from CT or MRI scans alone cannot be sufficiently accurate in such cases.

In the last decades, technology has led to significant advancements in image processing. 3D-reconstructions of CT images can improve surgical planning, allowing a careful evaluation of the hepatic lesions and their relationships with vascular and biliary structures. 3D-CT softwares are also able to calculate total liver volume (TLV) and the volume of each vessel's territories. In this way it is possible to evaluate FLR and plan the amount of liver that can be resected. This is essential in order to plan liver surgery.

## 3 Personalized Surgical Planning in Liver Surgery

First of all, it is necessary to analyze how the surgical planning in liver surgery is organized and how it is adapted to each patient. In this paragraph, we will deal with this topic. A preoperative evaluation of the patient is fundamental to plan the better treatment. When surgery is indicated, it is important to assess the future liver remnant (FLR) volume and function to plan the amount of liver to be resected. Anatomical study is essential to visualize the hepatic lesion and its relationship with hepatic arteries, hepatic veins, biliary tree, portal vein, lymph nodes and neighboring organs. Virtual 3D-models are very useful for this purpose and help surgeons to establish the correct surgical indication for each patient and perform a tailored surgery. In case of an insufficient FLR, it is also possible to generate a compensatory hypertrophy, allowing to resect liver tumors initially defined as unresectable.

### ***3.1 Preoperative Evaluation of the Patient***

One of the most important differences from other surgeries is that major liver resection requires a thorough preoperative evaluation with tailored surgical planning. It should be based on the evaluation of disease extension, histopathological examination, tumor growth pattern, vascular infiltration, tumor markers (e.g. AFP, CA 19.9, CEA), presence/absence of liver cirrhosis, presence/absence of fatty liver disease, liver volumetry, liver function, portal hypertension, resectability of the tumor, general health status of patient, risks related to each treatment option.

Treatment planning requires a multidisciplinary team, consisting of surgeons, medical oncologists, gastroenterologists, hepatologists, radiologists, interventional radiologists, anesthesiologist, intensivists, and pathologists [1].

Surgical treatment includes: wedge resection, segmentectomy, bisegmentectomy, hepatectomy, extended hepatectomy, super-enlarged hepatectomy, major hepatectomy, and simultaneous pancreatoduodenectomy.

### ***3.2 Future Liver Remnant***

In order to plan the surgery, it is necessary to evaluate the future liver remnant (FLR). It allows to increase the resectability rate of liver tumors in patients initially defined as unresectable. Despite this, the aim of preserving adequate liver remnant to avoid post-hepatectomy liver failure (PHLF) and a small for size syndrome represents the main limitation to obtain disease free margins (R0) after surgical treatment [2]. It is important to assess both the residual liver volume and function (i.e., quantity and quality, respectively).

For quantitative assessment, the preoperative volumetry of the liver can be performed using 3D-CT softwares, that are able to calculate total liver volume (TLV) and the volume of each vessel's territories. In this way it is possible to evaluate FLR. To be considered appropriate, it should be  $\geq 20\%$  of TLV in case of normal liver or  $\geq 30\%$  after chemotherapy. This percentage should be adjusted in patients with steatotic and cirrhotic liver according to the severity of the pathological condition [3].

Liver function should also be evaluated. It varies depending on whether the patient has a healthy liver, or hepatic steatosis, liver cirrhosis, chemotherapy associated liver injury (CALI). This evaluation can be done using scores such as Child–Pugh score or MELD score, liver stiffness with fibroelastogram, liver stiffness measurement (LSM), indocyanine green clearance (ICG).

### ***3.3 Anatomical Study***

Anatomical resection involves the removal of the entire portal territory that feeds the tumor, and is associated with better oncological outcomes. It is based on the Couinaud classification.

The anatomical study with 3D reconstruction allows a more accurate study of the hepatic lesion and intrahepatic vascular and biliary anatomy. It is possible to visualize the localization and course of hepatic arteries, hepatic veins, biliary tree, portal vein, lymph nodes, and relationship with neighboring organs [4].

### ***3.4 How to Increase the Remnant Liver Size***

Several strategies have been developed to generate a compensatory hypertrophy of the FLR, reducing the risk of post-hepatectomy liver failure (PHLF). Portal vein embolization is currently the standard of care. By reducing blood flow from the portal vein to one hemiliver, the other half of the liver is stimulated to grow, thereby causing it to enlarge. It increases the chances of performing extensive liver resection by inducing hypertrophy of the residual liver and reduces the risk of complications related to post-operative liver failure [5–7]. It is a procedure that is performed in an angiography room by an interventional radiologist. The time needed to induce hypertrophy is about four weeks. An abdominal CT scan with contrast is then performed to verify volumetric increase of the FLR, before performing surgery (major resection).

## **4 The Role of AI in Liver Surgery: Virtual 3D-Models**

In the last years, the advancement of AI-based technologies in medicine is becoming more and more important, although the majority of its real-world applications has not been implemented [8].

In liver surgery, we can see a practical application of AI. Surgery for liver tumors can be difficult due to intricate anatomy and worries about function of FLR. A deep understanding of liver anatomy is essential to perform a successful hepatic resection. Currently, in order to evaluate the position of a hepatic lesion, and its relationship with the surrounding structures in the preoperative planning, most HPB surgeons use two-dimensional (2D) images from computed tomography (CT) and magnetic resonance imaging (MRI) scans [9].

In some cases, mental reconstruction from CT or MRI scans alone can be challenging. Using virtual 3D-models in surgical planning is very useful in such cases. Two-dimensional images are not always sufficient to define the anatomical complexities of the liver and its vascular system. Four systems need to be carefully studied

in surgical planning for liver surgery: hepatic veins, hepatic arteries, biliary tree, and portal vein. Anomalies and variants of such systems should be identified.

Virtual 3D-models can provide precise anatomical details and spatial intrahepatic relationships. This is essential, especially in planning complex surgical procedures.

## 4.1 *Intricate Anatomy*

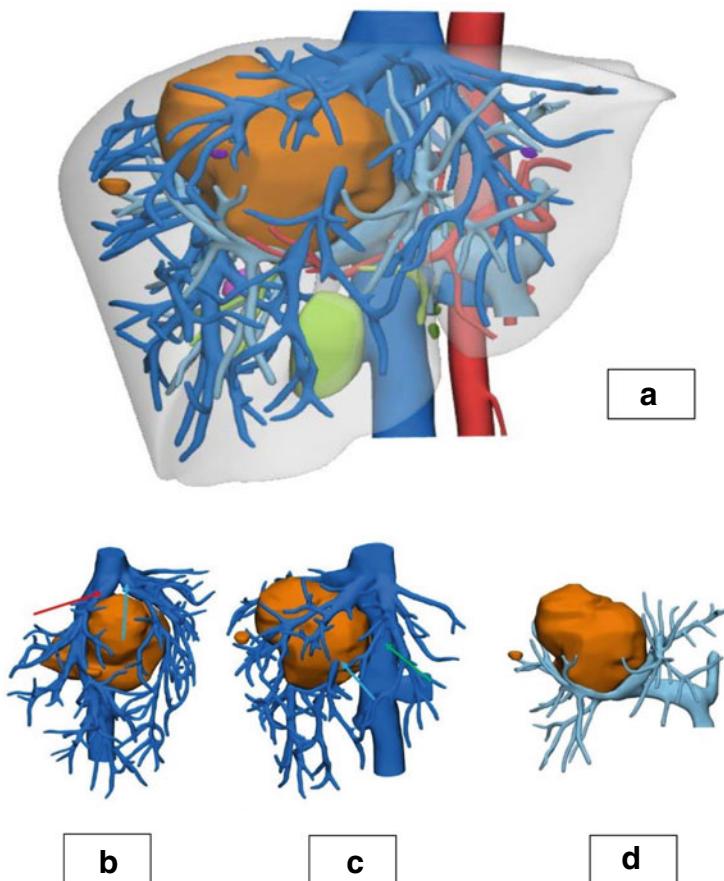
In some cases, having the 3D-reconstruction of the liver anatomy of our patients available is very useful. It allows us to carefully evaluate the hepatic lesion and its anatomical relationships.

In Fig. 1, the 3D-reconstruction of one of our patients' liver is showed. As we can see, the main lesion of about 11 cm, located in hepatic segments 5, 6, and 8, was in contact with the right hepatic vein, the middle hepatic vein, the retrohepatic vena cava, and with the bifurcation area of the portal vein in its main branches (Fig. 1a). In particular, virtual 3D-models allowed us to study carefully the relationship of the lesion with the hepatic veins (Fig. 1b, c) and with the branches of the portal vein (Fig. 1d).

The careful evaluation of the patient's anatomy allowed us to plan and perform a tailored surgery. A surgical exeresis of the neoformation was performed. The surgical approach required a luxation of the liver. With this approach, an almost total hepatic vascular exclusion was determined due to torsion of both the vena cava and the portal vein. In this postero-anterior parenchymal approach, it was possible to isolate the neoformation respecting the vascular structures (hepatic veins, portal vein branches, and hepatic arteries) and biliary structures, dissecting only the few afferent and efferent branches of the mass. The maneuver was practiced intermittently. It was performed for periods of ten minutes with recovery intervals of five minutes, to improve tolerance to hepatic ischemia, allowing the increase in cumulative ischemia time and achieving the objective of exeresis of the neoformation.

## 4.2 *Identifying Biliary Branches and Vascular Structures to Dissect*

Virtual 3D-models allow us to identify which biliary and vascular branches are affected. This is important in choosing which biliary branches and vascular structures to dissect and at what level. The amount of liver that will need to be removed will also depend on this choice. A careful preoperative evaluation therefore can allow us to preserve more residual liver. Parenchyma-sparing liver surgery is fundamental to reduce the risk of complications, as post-hepatectomy liver failure [10].



**Fig. 1** 3D-reconstruction of a patient's liver (**a**) and Relationship of the lesion with the hepatic veins (**b** and **c**) and Branches of the portal vein (**d**). Study and realization by Medical Device Custom Made with HA3D™ technology (Hyper accuracy 3D™). Provided by MEDICS srl—Torino, Italia

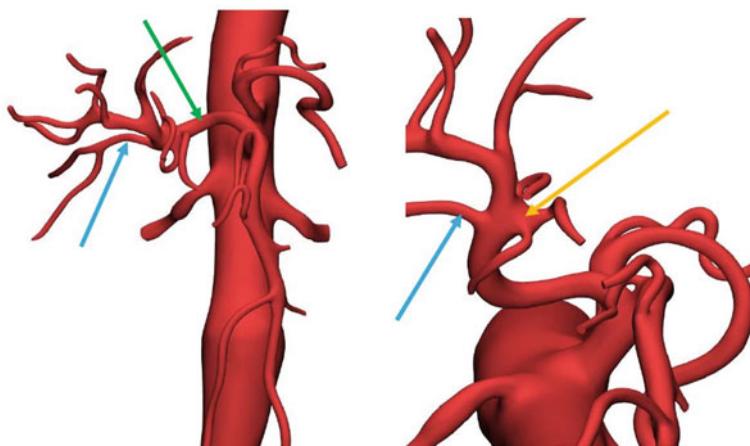
#### **4.3 Establishing the Correct Surgical Indication for Each Patient**

3D-reconstructions help us to establish the correct surgical indication for each patient. The preoperative evaluation of the affected biliary branches and vascular structures can allow radical interventions, ensuring R0 margins. If a radical intervention cannot be performed, it allows us to evaluate with greater precision whether it is better to consider the tumor unresectable or to perform a non-radical intervention. These options will clearly be accompanied by the evaluation of the possibility of undertaking non-surgical options, such as chemotherapy.

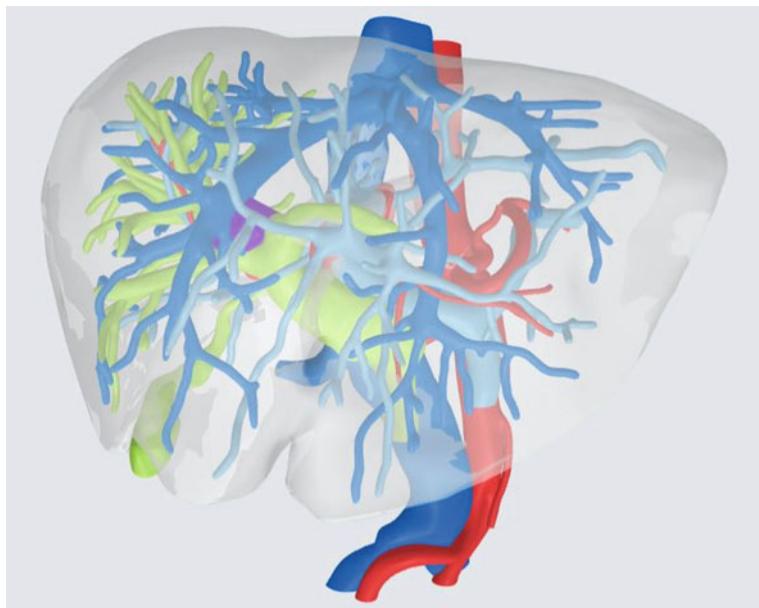
3D-models allow us to better plan the amount of liver to be resected, evaluate whether or not surgery is indicated, and whether or not procedures to increase FLR such as portal vein embolization, are indicated.

#### 4.4 Anomalies and Variants

3D-reconstructions are also very useful to visualize anatomical anomalies and variants. In Fig. 2, we can see the 3D-reconstruction of the vascular anatomy of a patient with some anomalies. The proper hepatic artery arises from the superior mesenteric artery. The right posterior arteries arise independently directly from the proper hepatic artery. The gastroduodenal artery appears very short and forks early into the right gastroepiploic and superior pancreaticoduodenal arteries. This information can be very useful depending on the surgical procedure that we are planning to perform. It is not uncommon to detect the presence of anatomical variations of the hepatic artery. It is very important to identify these variations and evaluate whether they are aberrant or accessory hepatic arteries.



**Fig. 2** 3D-reconstruction of vascular anatomy of a patient with anatomical anomalies. Green arrow: proper hepatic artery; Blue arrow: right posterior arteries; Yellow arrow: gastroduodenal artery. Study and realization by Medical Device Custom Made with HA3D™ technology (Hyper accuracy 3D™). Provided by MEDICS srl—Torino, Italia



**Fig. 3** 3D-reconstruction of a liver of a patient with cholangiocarcinoma, who underwent hepatopancreatoduodenectomy. Study and realization by Medical Device Custom Made with HA3D™ technology (Hyper accuracy 3D™). Provided by MEDICS srl—Torino, Italia

#### 4.5 Adequate FLR

In case of extended liver resection, it is very important to carefully evaluate the FLR. In Fig. 3, we can see the 3D-reconstruction of a liver of a patient with cholangiocarcinoma. Surgical resection with negative margins (R0) associated with regional lymphadenectomy is currently the only potentially curative treatment in patients with resectable cholangiocarcinoma. The patient underwent hepatopancreatoduodenectomy. The role of this surgical procedure is still controversial due to high morbidity and mortality rates, but sometimes it is the only potentially curative treatment for the patient. In such cases, preoperative assessment is fundamental to identify patients that could benefit from this complex surgical procedure.

### 5 Future Perspectives and Challenges

AI stimulates the advancement of liver surgery. 3D-reconstructions of CT images can improve surgical planning in selected cases. Visualizing the spatial relationship of the hepatic lesion and surrounding structures, identifying the normal vascular and

biliary anatomy and its variations, as well as evaluating FLV are fundamental in liver surgery.

Current applications in this field will be surely implemented in the next years. Technological advances will allow increasingly detailed reconstructions and more precise estimates of FLV, improving surgical outcomes and reducing the risk of post-operative complications.

Virtual 3D-models are very useful for HPB surgeons, but currently 3D-CT softwares are used in a few centers. Centers that do not have these softwares can send the CT images to such centers to perform these studies, but there are clearly costs. Despite all the advantages abovementioned of the virtual 3D-models in surgical planning of liver surgery, this is the major limitation.

In future it would be beneficial for every center, where liver surgery is performed, to be equipped with such software.

## 6 Conclusions

Liver surgery requires in-depth personalized preoperative planning and a multidisciplinary team. Study of residual liver volume and function is imperative in case of major liver resections. FLR should be adequate to avoid complications, as post-hepatectomy liver failure. Procedures to increase the FLR size, as portal vein embolization, can be performed, increasing the resectability rate of liver tumors in patients initially defined as unresectable.

Anatomical study is mandatory. 3D-reconstructions of 2D images from CT scans can really help the surgeon to visualize the spatial relationship of the hepatic lesion and surrounding intrahepatic structures. They can also allow the surgeon to identify the normal vascular and biliary anatomy and its anomalies and variations, improving preoperative surgical planning. After this careful evaluation, some patients considered unresectable on 2D images can become resectable or vice versa. 3D-models allow us to better plan the amount of liver to be resected and establish the correct surgical indication for each patient.

## References

1. Maki H, Hasegawa K (2022) Advances in the surgical treatment of liver cancer. *Biosci Trends* 16(3):178–188. <https://doi.org/10.5582/bst.2022.01245>
2. Cassese G, Han HS, Al Farai A, Guiu B, Troisi RI, Panaro F (2022) Future remnant liver optimization: preoperative assessment, volume augmentation procedures and management of PVE failure. *Minerva Surg* 77(4):368–379. <https://doi.org/10.23736/S2724-5691.22.09541-7>
3. Karanicolas PJ (2017) Assessment of hepatic function: implications for the surgical patient. In: Jarnagin WR, Allen PJ et al (eds) Blumgart's surgery of the liver, biliary tract, and pancreas. Elsevier, Philadelphia, pp 60–65

4. Liu JP, Lerut J, Yang Z, Li ZK, Zheng SS (2022) Three-dimensional modeling in complex liver surgery and liver transplantation. *Hepatobiliary Pancreat Dis Int* 21(4):318–324. <https://doi.org/10.1016/j.hbpd.2022.05.012>
5. Shindoh J, Madoff DC, Aloia TA, Vauthey JN (2017) Preoperative portal vein embolization: technique and results. In: Jarnagin WR, Allen PJ et al (eds) Blumgart's surgery of the liver, biliary tract, and pancreas. Elsevier, Philadelphia, pp 1653–1663
6. Kokudo N, Takemura N, Ito K, Mihara F (2020) The history of liver surgery: achievements over the past 50 years. *Ann Gastroenterol Surg* 4(2):109–117. PMID: 32258975; PMCID: PMC7105847. <https://doi.org/10.1002/agrs.12322>
7. Kishi Y, Vauthey JN (2021) Issues to be considered to address the future liver remnant prior to major hepatectomy. *Surg Today* 51(4):472–484. <https://doi.org/10.1007/s00595-020-02088-2>
8. Taha A, Ochs V, Kayhan LN, Enodien B, Frey DM, Krähenbühl L, Taha-Mehlitz S (2022) Advancements of artificial intelligence in liver-associated diseases and surgery. *Medicina* 58(4):459. <https://doi.org/10.3390/medicina58040459>
9. Bari H, Wadhwani S, Dasari BVM (2021) Role of artificial intelligence in hepatobiliary and pancreatic surgery. *World J Gastrointest Surg* 13(1):7–18. <https://doi.org/10.4240/wjgs.v13.i1.7>
10. Winner M, Pawlik TM (2017) Parenchymal preservation in hepatic resectional surgery: rationale and indications. In: Jarnagin WR, Allen PJ et al (eds) Blumgart's Surgery of the liver, biliary tract, and pancreas. Elsevier, Philadelphia, pp 1631–1643

# Impact of AI on Student's Research and Writing Projects



Joan Rosselló-Geli

**Abstract** In recent years, the field of artificial intelligence (AI) has grown rapidly and has revolutionized various areas of life. One of these areas is research, where AI is used to enhance and streamline the research process. Students are among those who have benefited from the use of AI in research, as it has made the process more efficient and effective. The use of AI applications to improve essays and reports is a growing trend that needs being taken into account by professors in order to assess properly the student's contributions. A large concern affects faculties worldwide regarding the possibility of cheating. How it will influence the educational process is another question. Changes about how assess and how to teach are needed or the trend will end somehow and somewhat? In that sense, we will explore the impact of AI on student research, and how it has changed the way students approach writing reports. Finally, we will focus on how professorship has modified the way to assess students work.

**Keywords** Student's reports · AI applications · Writing · Research

## 1 Introduction

Artificial intelligence can be defined as a computer program trying to imitate intelligent behavior. In that sense, AI is a system that imitates intelligent human behavior [1]. However, there is not a widely accepted definition of what AI is, as the complexity of defining intelligence makes it difficult [2].

In recent years, the field of AI has grown rapidly and has revolutionized various areas of life. One of these areas is research, in which AI is being used to enhance and streamline the research process [3]. In that sense, students are among those who have benefited from the use of AI in research, as it has made the process more efficient and effective, showing the high potential of AI for educational purposes [4].

---

J. Rosselló-Geli (✉)  
Universitat Oberta de Catalunya, 08018 Barcelona, Spain  
e-mail: [jrosseloge@uoc.edu](mailto:jrosseloge@uoc.edu)

There is a large number of impacts of AI on student research:

- **Improved Search Capabilities.** One of the most significant impacts of AI on student research is the improvement of search capabilities. AI-powered search engines, such as Google Scholar and Microsoft Academic, allow students to find relevant research papers and articles more quickly and easily. These search engines use algorithms that analyze the content of research papers and articles and provide accurate results based on the student's search query. This has allowed students to find relevant research more quickly, thus saving time and effort.
- **Rapid Data Analysis.** Another impact of AI on student research is the ability to analyze wide amounts of data quickly and accurately. This is particularly useful for students who are conducting research that involves analyzing large data sets. AI-powered tools, such as RapidMiner, can analyze data and provide insights that would be difficult or impossible for humans to develop on their own. This has facilitated to make sense of complex data and draw meaningful conclusions from their research.
- **Improved Accuracy.** AI has also improved the accuracy of research. AI helps to identify errors and inconsistencies in research papers and articles, ensuring that students are basing their research on valid information. This has led to process that is more reliable and has helped students to avoid making mistakes that could undermine their research findings.
- **Improved Writing.** Writing tools, such as Grammarly and Hemingway, have also had a significant impact. These applications can analyze a student's writing and provide suggestions for improvements, such as grammar and spelling corrections, as well as recommendations for more concise and effective writing. This has helped students to ameliorate the quality of their compositions and communicate their research findings more effectively.
- **Collaboration.** Finally, AI has also facilitated collaboration among students conducting research. AI-powered collaboration tools, like Slack and Trello, permit to work together more efficiently and effectively. These tools enable them to share research papers and articles, discuss research findings, and coordinate their efforts in real-time. This has resulted in the production of high-quality research papers and articles.

Nevertheless, AI applications require a responsible use in research, as there is a risk of misuse and possible abuses of the available data [5–8]. In that sense, a current threat related to the use of AI in education is the possibility of cheating while writing research assignments. Plagiarism is an important issue that needs to be addressed to maintain academic honesty [9]. Although it is not a new occurrence [10], an increase of plagiarism has been detected among university students lately, even if a proper use of technology can reduce it [11]. In Spain, 61% of students admitted having committed some form of plagiarism, mainly using online texts without including a proper citation [12].

Such results can also be found within secondary student's research projects. The use of Internet to look for information and the large amount of available resources leads to an easy way to gather and copy information, which is included in the students' writings without citing the source, leading to the so-called "copy and paste" generation [13–15], a generation that makes the same mistakes once they start their university studies.

All of the above is of great concern to both academic institutions and professors. To tackle the plagiarism issue, there are strategies such as the use of text matching software [16], which is widely available at present, as well as the penalization of students who are caught plagiarizing. In contrast, other authors suggest that what is necessary is to increase awareness to ensure the honesty of its work, thus promoting academic integrity [17, 18].

The chapter herein presented deals with the impact of AI on cheating while trying to avoid detection and the proposed solutions both from the academic authorities and by the lecturer in his class.

## 2 The UOC and Its e-Learning System

The Universitat Oberta de Catalunya (UOC) was born in 1995 with 200 students and 2 majors. In the current school year, it exceeds 60,000 students and offers more than 20 degrees as well as masters, language courses, and doctorates [19].

UOC was the first worldwide exclusively online university. Its focus has always been the development of an educational model that incentives research and innovation through e-learning, allowing a high-quality education for everyone, everywhere.

Their methodology is based on a customized teaching support combined with activities, which allow each student to set their own path toward graduation. Each semester, students opt for some subjects and must develop reading activities from different sources (UOC own material, scientific papers, books) ending with an essay that is compulsory to pass the subject. At the end of the semester, students face a writing exam, which differs for those having completed the essays and those selecting to do a single test, an exam of all the contents of the semester.

Before COVID-19, students took the final exams at the university itself in Barcelona or in the university's buildings distributed in the different Spanish regions and in some European capitals. In that sense, it was easier to avoid problems, such as false identities or copying while testing but the pandemic stopped such exams, moving them online.

The worldwide trend toward teaching and assessing online caused an increase of cheating and the search for information on how to cheat during online examinations, as happened in Spain [20]. To avoid such fraudulent activities, the UOC imposed a system in which the student, obligatorily, had to be taking the test with the computer camera on and could not get up or make any suspicious movements during the duration of the exam. The computer system alerts the correcting teacher of any incident, and this may result in failure of the exam.

The examination system was retained once the lockdown affecting Spain was over, but the face-to-face activities were kept at a minimum for a long period. Nowadays, the system is fully online, and on-site examinations will not be retaken.

### 3 AI as a Cheating Tool

Although as mentioned above plagiarism using AI is not new, there has been a recent increase of concern as new AI tools became widely accessible online. In that sense, the arrival of chatbots allowing students to generate academic works without the fear of being caught has generated a discussion around the educational system, from secondary schools to the university.

A chatbot is a program that simulates human-like conversations in a conversational manner [21]. At the end of November 2022, a new AI chatbot was released with the commercial name of ChatGPT. Less than a week later, the firsts articles and reports about its use as a plagiarism tool arrived to the press [9] and concerns rapidly increased. Moreover, some authors predicted that such application will change the way to assess college work [22, 23].

The impact of bots like ChatGPT is larger than the impact provoked by other AI applications like the writing assistants. Applications such as Grammarly affect the way students write, in some cases improving while in others worsening the quality of the work [24].

The improvement of the AI capabilities, together with the increased possibilities of access to its use, offers students a wide range of options. Selecting the positive ones is a joint work between lecturers and students, in addition to the academic institutions. On the other hand, the choice to make a fraudulent use may be caused by a lack of awareness of the consequences of their actions, as well as a feeling of impunity in the belief that they cannot be caught. Such behavior is common for students using contract-cheating tools, either by means of the so-called “essay mills”, or by means of outsourcing assignments to friends or other students [25].

### 4 Answers to AI Impact

The UOC implemented a system to detect cheating on academic essays and reports developed by the students during the personalized part of work of the semester. An anti-plagiarism system is available to professors, as each classroom has a link to the PAC-plagi Website, where statistics for all the semester essays and activities are kept. Furthermore, the data allows them to know the percentage of copy present on each student’s work (Fig. 1).

Mostra: 25	1	2	39	Mostrando (1-25) de 39
Estudiant	Percentatge de coincidència amb altres estudiants		Percentatge de coincidència amb fonts externes	
	7 %		3 %	
	0 %		11 %	
	6 %		16 %	
	4 %		7 %	
	4 %		29 %	
	2 %		11 %	
	2 %		7 %	
	2 %		20 %	
	2 %		6 %	
	2 %		-	
	1 %		4 %	
	1 %		11 %	
	1 %		3 %	

**Fig. 1** Snapshot of the plagiarism percentage information

The first column includes the student identification data, while the second column states the percentage of copy related to other students essays. Finally, the third column informs of the percentage of plagiarism from external sources, that is, from Internet-obtained information.

The data related to other students is important as, in some occasions, there has been an exchange of work between students, especially if the same questions are asked in consecutive semesters.

The smallest percentages are often related to the copy of titles and bibliographical resources and once checked are accepted. On the other hand, values over 25% are considered intentional, and the students are notified accordingly.

In order to help students to avoid unintentional plagiarism, resources explaining how to cite and how to avoid plagiarism mistakes are available at the university library.

As stated before, during the COVID-19 pandemic, there was the need to change the way final examinations which were undertaken. It was impossible to do in-person exams, as there was no available space to accommodate so many students. The new online system has flaws, especially in cases of power failures and network outages. Students receive the questions and have a time to answer and upload the copy to the server. If they face technical difficulties, they have a phone number to call and report problems and ask for help. All the written examinations are checked for plagiarism, and a same report as shown in Fig. 1 is available to the lecturers.

Starting the past school year, some degrees began to suggest to their first year students the possibility of developing a previous writing essay, without a grade, to exercise in the use of quotations as well as to learn the different types of plagiarism. The initial results show that students who carry out this activity have lower plagiarism percentages during the academic year.

**Fig. 2** Infographic on how to prevent academic misconduct. Source [26]



At the beginning of the current academic year, the main concern among teachers is the possible use of IA applications, which cannot be detected by anti-plagiarism tools. This concern, conveyed to university authorities, has resulted in the creation of infographics to help professors tackle this problem (Fig. 2) and, in addition, to help them explain to students how these tools can be used in an appropriate and useful way (Fig. 3).

In addition to educating students, lecturers should develop their own tools to detect whether students have used inappropriate software. In this sense, the use of thought-provoking questions and critical analysis of texts can be basic to uncover attempts at deception [27]. Another option is the use of formative assessments, thus allowing students to reckon and accept their mistakes without the fear of being penalized with lower grades [28].

The infographic is titled "How to explain to students what they can do using AI\*" and is part of a series titled "Use of generative AI in classrooms". It features five numbered points (1 to 5) with icons and explanatory text. A sidebar on the left asks if AI tools like ChatGPT or Copilot are allowed in assessments, with a note that they must be cited if used.

**Can you use a generative AI tool (ChatGPT, Bing, Copilot, etc.) in a continuous assessment activity?**

**1 ✗**  
In general, you are not allowed to use generative AI tools to achieve the main goal of a given assessment activity. For instance, using Copilot in introductory programming courses or ChatGPT in written communication skills courses is not allowed.

**2 ✓**  
You should always check with the course instructor whether the use of a particular tool is allowed for a given assessment activity. In particular, you are not allowed to use generative AI tools in final tests or exams unless it is explicitly permitted.

**3 ⚡**  
When the use of generative AI tools is allowed, they must be properly cited in your submission, in the same way that you must cite other external resources that have been used to solve an assessment activity.

**4 🚫**  
The use of generative AI assistants should be limited in scope (minor edits, fine-tuning solutions, etc.). If a plagiarism detection service flags your submission as suspicious, this is a warning sign of an abuse of generative AI, which could have an impact on your grade.

**5 ⚠**  
You are always responsible for the content of your submissions for assessment activities. This means that any error appearing in your work delivery will be attributed to you as the author. Similarly, you should understand all the details contained in your submission and be able to explain them or justify the decisions taken; if you cannot, your work delivery may be given a fail grade, even if it is otherwise correct.

(\*) Indications developed by teachers Robert Clarisó y Toni Pérez del EIIMT.

eLearning Innovation Center    UOC Universitat Oberta de Catalunya

eLearning Innovation Center - UOC 2023

Fig. 3 Infographic to explain to students how to use correctly AI tools. Source [26]

## 5 Conclusion

The arrival of AI has had a significant impact on student work. It has improved search capabilities, data analysis, accuracy, writing, and collaboration. These benefits have made it easier for students to conduct research, produce high-quality research papers and articles, and communicate their research findings effectively. As AI technology continues to evolve, we can expect that it will have an even greater impact on student research in future.

On the other hand, while AI can provide many benefits for students, it is essential to monitor its use to prevent cheating. Professors and institutions should have policies in place to detect and prevent cheating and ensure that students are learning and demonstrating their skills honestly.

In that sense, the improvement of how academic honesty is perceived and instilled among students, from the early stages, is basic to the correct use of AI applications. Since last semester, the introductory subjects of the Geography and History course are developing an activity of correct citation and plagiarism prevention, which complements the actions proposed by the university itself, tending to penalize plagiarism by developing applications that allow its discovery both in written assignments and in exams. In spite of being in an initial stage and not having quantitative values, the first results are very positive, insofar as citations have improved, and plagiarism percentages have been reduced in the work assigned to students.

Finally, the development of AI tools capable of writing essays and other academic works, such as ChatGPT or others, has provoked the need to rethink the way of evaluating the personal work of each student before reaching the final examinations. In this sense, changes can be foreseen in the near future, both in essays and in virtual tests, given the difficulty of controlling the use of these AI tools and the ease with which plagiarism can occur.

## References

1. Kok JN, Boers EJW, Kosters WA, Van der Putten P, Poel M. Artificial intelligence: definition, trends, techniques and cases. In: Artificial intelligence, vol 1. Encyclopedia of Life Support Systems (EOLSS). UNESCO-EOLSS.
2. Wang P (2019) On defining artificial intelligence. *J Artif Gen Intell* 10(2):1–37
3. Chassignol M, Khoroshavin A, Klimova A, Bilyatdinova A (2018) Artificial Intelligence trends in education: a narrative overview. *Procedia Comput Sci* 136:16–24
4. Holmes W, Tuomi I (2022) State of art and practice in AI in education. *Eur J Educ* 57(4):542–570
5. Zanetti M, Pendina S, Piceci L, Cassese FP (2020) Potential risks of artificial intelligence in education. *Form@re* 20(1):3687
6. Schiff D (2021) Out of the laboratory and into the classroom: the future of artificial intelligence in education. *AI Soc* 36(1):331–348
7. Klimova B, Pikhart M, Kacatl J (2023) Ethical issues of the use of AI-driven mobile apps for education. *Front Public Health* 10:1118116
8. Humble N, Mozelius P (2022) The threat, hype and promise of artificial intelligence in education. *Discov Artif Intell* 2:22

9. Cotton DRE, Cotton PA, Shipway JR (2023) Chatting and cheating: ensuring academic integrity in the era of ChatGPT. *Innov Educ Teach Int* 1–12
10. Hart M, Friesner T (2004) Plagiarism and poor academic practice-a threat to the extension of e-learning in higher education? *Electron J e-Learn* 2(1):89–96
11. Torres-Díaz JC, Duart JM, Hinojosa-Becerra M (2018) Plagiarism, internet and academic success at the university. *J New Approaches Educ Res* 7(2):98–104
12. Egaña T (2012) Uso de bibliografía y plagio académico entre los estudiantes universitarios. *Rev Univ Soc Conocimiento* 9(2):18–30
13. Sureda J, Comas R, Urbina S (2005) The “copy and paste” generation: plagiarism amongst students, a review of existing literature. *Int J Learn* 12
14. Inclán C (2016) Ctrl-C, Ctrl-V. La práctica escolar de copiar y pegar en el bachillerato. *Perfiles Educativos* 38(154):6–11
15. Molina Salinas JA (2018) El copiar y pegar ¿nueva estrategia de aprendizaje? *Apuntes Ciencias Sociales* 8(2):179–186
16. Bretag T (2013) Challenges in addressing plagiarism in education. *PLoS Med* 10(12):e1001574
17. Trobey K, Roach D, Cochran L (2005) Encouraging academic honesty in student work. *J Bus Adm Online* 4(2)
18. Cebrián Robles V, Raposo Rivas V, Sarmiento Campos JA (2020) Study of the reasons for and measures to avoid plagiarism in young students of education. *Profesorado* 24(1):50–74
19. UOC Nuestra historia. <https://www.uoc.edu/portal/es/25-anys/historia/index.html>
20. Comas-Forgas R, Lancaster T, Calvo-Sastre A, Sureda-Negre J (2021) Exam cheating and academic integrity breaches during the COVID-19 pandemic: an analysis of internet search activity in Spain. *Helyon* 7(10):e08233
21. Khalil M, Er E (2023) Will ChatGPT get you caught? Rethinking of plagiarism detection. [arXiv:2302.04335](https://arxiv.org/abs/2302.04335)
22. Stokel-Walker C (2022) AI bot ChatGPT writes smart essays—should professors worry? *Nature*. D41586-022-04397-7
23. Yeadon W, Inyang OO, Mizouri A, Peach A, Testrow C (2022) The death of the short-form physics essay in the coming AI revolution. [arXiv preprint arxiv:2212.11661](https://arxiv.org/abs/2212.11661)
24. Papakonstantinidis S. AI writing assistants in higher education: a mixed blessing? *LinkedIn*. <http://linkedin.com/pulse/>
25. Harper R, Bretag T, Rundle K (2021) Detecting contract cheating: examining the role of assessment type. *High Educ Res Dev* 40(2):263–278
26. UOC e-Learning Innovation Center: e-Learning Kit (2023)
27. Surahman E, Wang T-H (2022) Academic dishonesty and trustworthy assessment in online learning: a systematic literature review. *J Comput Assist Learn* 38(6):1535–1553
28. Leung CH, Ling Cheng SC (2017) An instructional approach to practical solutions for plagiarism. *Univ J Educ Res* 5(9):1646–1652

# Spaces, Videogames and Artificial Intelligence: A Geographical Approach



Gaetano Sabato and Francesco De Pascale

**Abstract** Artificial intelligence (AI) is a branch of computer science that deals with creating machines capable of imitating human intelligence. This is accomplished through the programming of complex algorithms, data collection and analysis or the use of neural networks. When it comes to the relationship between video games and geography, AI can be used to create more realistic and interactive game environments and, in general, it involves different forms of spatialities (virtual and symbolic). Additionally, AI can be used to tailor the gameplay experience to the player's geographic location, offering specific content and challenges. From the perspective of cultural geography, the paper aims to propose some reflections on how AI can play an important role in cultural geography and create more spatialized, engaging and personalized gaming experiences able to integrate elements of real and virtual geography.

**Keywords** AI · Cultural studies · Geography · Spaces · Video games

## 1 Introduction

Artificial intelligence<sup>1</sup> (AI) is a discipline belonging to Information Technology, a discipline that studies methods and tools that create machines to do things that appear intelligent. The first studies on modern AI date back to 1943 with an article describing how an artificial neuron could work [1]. The name AI dates back to 1955, to a proposal

---

<sup>1</sup> The authors share the design and drafting of this paper. However, Francesco De Pascale is the author of Sects. 1 and 2. Gaetano Sabato is the author of Sects. 3 and 4.

G. Sabato (✉)

Department of Psychological, Pedagogical, Physical Exercise and Training Sciences, University of Palermo, Palermo, Italy

e-mail: [gaetano.sabato@unipa.it](mailto:gaetano.sabato@unipa.it)

F. De Pascale

Department of Communication Sciences, University of Teramo, Teramo, Italy  
e-mail: [fdepascale@unite.it](mailto:fdepascale@unite.it)

for a summer conference held in 1956 in Dartmouth, and from this time there has been a series of periods of enthusiasm for a single paradigm, followed by periods of disappointment with the limitations subsequently encountered. Currently, we are in a new period of enthusiasm, largely as the result of the use of simulated neural networks that contain many layers and connections [2]. Contemporary research aims to integrate various methods that have emerged over time in order to overcome the limitations produced by previous paradigms.

In 1950 Alan Mathison Turing contributed an article to the journal *Mind* entitled “Computing Machinery and Intelligence”, in which he pondered whether a machine could ever think [3]. A large part of subsequent studies on AI are based on his article. In the article, Turing argued that to answer this question he should be able to define the terms “machine” and “thought” [3]. Regarding the first term, Turing managed to describe an ideal machine capable of performing any type of calculation which still, currently, forms the basis of the theoretical description of all our computers: the so-called “Turing Machine” [3]. Faced with the problem of defining thought, however, Turing resorted to an operating method, known as the “Turing Test” or “imitation game”, which claimed that if after a certain time we cannot distinguish the machine from human thought we can say that the machine is smart. The Turing Test is used to this day to distinguish human intelligence from artificial intelligence [3].

The importance of relations between geography and AI can be contextualized by considering the 17 sustainable development goals (SDGs) defined by the United Nations in their Agenda 2030. Environmental and computing challenges will affect the well-being of mankind, sustainable development, and resilience of the entire planet [2]. According to Ferilli, AI can make a concrete contribution to sustainable development, with systems capable of measuring baseline conditions and how far we are from achieving each objective; how to optimize available resources; analyze the impacts of potential strategies and suggest innovative solutions. However, the need to manage risks associated with the use of AI remains fundamental [2]. First of all, we need to consider AI’s current environmental impact. To train neural networks and machine learning algorithms, on which AI depends, a significant amount of energy is required and is mostly generated by fossil fuels. A geographic perspective can help identify key factors affecting who gains and who loses in the face of AI and how its technological transformations are intertwined and impact the spatial configuration of the capitalist economy [4]. AI in general, and deep learning in particular, have entered scientific debates on how to address many geographical problems, from the detection of topologies to satellite imagery [5–8] to urban traffic prediction [9, 10], or the recent trend of GeoAI development within the GIScience discipline [5, 11–14] and the relationship between spaces, videogames and AI [15]. Within this context, this paper will discuss connections between AI, video games, human geography and concepts and configurations of space.

## 2 Geographies and AI

AI is driving significant new opportunities and challenges for geospatial research. Its rapid development is fuelled by theoretical advances, big data, computer hardware (e.g. graphics processing units or GPUs), and high-performance computing platforms that foster intelligence development, training and artificial modelling [16]. Vopham et al. [17] define GeoAI as “an emerging scientific discipline that combines innovations in spatial science, artificial intelligence methods in machine learning (e.g. deep learning), data mining and high-performance computing to extract knowledge from spatial big data” and highlights that “the moulding together of artificial intelligence (AI) and the geographic information systems (GIS) dimension creates GeoAI” [18]. In urban geography, conventional GeoAI methods have made it possible to address and analyze many of the urban problems of recent years. However, while the data location information has been applied to spatial data, it has not been used directly by the models, e.g. unsupervised clustering methods in geodemographic classifications do not use information about the geospatial context of an area. This means that conventional models do not take into account the influence of geospatial context on data distribution. Integrating geospatial context information could lead to greater accuracy in modelling urban problems [5, 19–23].

Another important insight concerns the relationship between AI and behavioural geography [24]. An initial goal was to deploy AI to replace the daily, time-consuming or labour-intensive tasks of geographic analysis. Smith [25] presents two approaches to the use of AI in geography: the engineering approach, which considers AI as a set of machine procedures to perform tasks and the cognitive approach, which proposes AI as a mimetic means to represent human processes of analytical intelligence. In summary, Smith’s thinking on AI in geography focusses on Simon’s invocation of “artificial thinking” as a means to bridge the gap between engineering and cognitive AI analysis and conceptualization [26]. This involves the use of machines that show similarity of process and product to geographers, with the possibility of taking over some of their analytical skills and tasks [27]. Smith lists several examples of geographical activities that could be performed by machines, such as acquiring knowledge, organizing and reasoning about decisions [24]. Concerning the relationship between AI and research in economic geography, Cicerone et al. [28] include patent-based innovation studies to investigate how AI can be used for the development of green technologies at the regional level. The authors propose to broaden the analysis to the ability of regions to acquire and maintain a green-tech specialization over time, focussing on potential GPT technologies (technologies with general purpose properties) such as AI. The authors believe that AI can be considered a potential “invention in the method of inventing” which requires an original recombination of a fairly large set of data, collated and analyzed at the local level. Therefore, the authors believe that knowledge of green technology in any given region is crucial in considering the local AI knowledge stock.

Li and Hsu [29] discuss the evolution of human geography into a field of big data science, highlighting the use of global observation systems and distributed

sensor networks in cities to collect vast amounts of geospatial information. For social applications involving neo geographic technologies [30], the prevalent use of location-based social media, GPS-enabled handheld devices, various volunteer geographic information (VGI) platforms, and other “social sensors” have fostered the creation of massive datasets on mobility intelligence, public opinion and human carbon footprints on a large scale. This information, coming from various sources, is in different formats and contains a wide range of data. This presents a challenge to traditional analytical methods, but also offers the possibility of discovering new geospatial knowledge through the use of artificial intelligence and machine learning. In summary, data-driven geography has become a new method to better analyze and understand territorial ecosystems.

AI can also be of great help to the branches of risk geography and disaster studies. Kuglitsch et al. [31] discuss the potential of AI in the management of disasters related to natural hazards, such as earthquakes, floods, landslides and forest fires. AI can help prevent and mitigate damage through disaster prediction, identification of risk areas, coordination of emergency responses and post-event damage assessment. Their study also explores the limitations of AI in disaster management, such as reliance on data sources and the need for strong technology infrastructures. However, AI can still provide significant value in disaster management, particularly through the integration of geospatial data and machine learning. Therefore, AI can be a useful tool in disaster management, but its effectiveness depends on the quality of the data and the available technological infrastructure.

The use of AI is becoming increasingly prevalent in various fields and has the potential to impact global productivity, equality, inclusion and environmental outcomes. While AI can positively impact sustainable development, there is currently no comprehensive study on the extent to which it could affect all aspects of the United Nations Agenda 2030 Sustainable Development Goals and their 169 internationally agreed goals. However, it is clear that AI can both positively and negatively impact the ability to achieve these goals [32]. Vinuesa et al. [32] explore the role of AI in achieving these United Nations Sustainable Development Goals. Using an expert elicitation process, it was illustrated that AI can contribute to the achievement of 134 goals, but also inhibit 59 goals. However, the rapid development of AI requires increased knowledge and regulatory oversight to ensure AI-based technologies are ethical and safe. Failure to do so may result in gaps in transparency and ethical standards, undermining long-term sustainability. AI has important epistemological, methodological and ethical implications that must be addressed to achieve the key challenges and actions for the future of the planet; the connections and ways of interaction between geography and AI can help us understand and reflect in a complex and integrated way on these topics and challenges.

### 3 Video Games, Spaces and AI

For more than thirty years, video games have been an established and growing form of entertainment. During this period, they have become increasingly refined and diversified and like with other cultural products, genres and sub-genres can be distinguished. Obviously, the development of this industry is closely linked to that of information technology and digital and online platforms: indeed, the most utilized game machines are mobile devices, PCs and consoles. According to recent data from the annual report conducted in 36 countries [33] on all continents, it is estimated that in 2022 video gamers worldwide were 3228 million, while the previous year they were 3079 million. Most come from the Asia–Pacific area (54%), then from Middle East and Africa (16%), Europe (13%), Latin America (10%) and North America (7%). Considering video gamers by platform, 2698 million were mobile players, 1054 million PC players and 611 million console players. The gaming market at is worth a lot in economic terms: suffice it to say that according to the same report game sales generated more than 184 billion dollars [33]. Even COVID-19's lockdown measures contributed to this rapid growth in the last three years [34]: in many countries the prolonged periods of confinement have brought many people closer to video games as a form of home entertainment. Video games have long been a privileged object of study for the social sciences. This interest has given rise to a vast interdisciplinary literature that often flows into games studies as part of cultural studies [15]. Among others, studies in the cultural anthropological [35, 36], historical [37], psychological [38], sociological [39, 40], politological [41–43], narratological [44–46] fields can be included. From the perspective of cultural geography video games are an interesting subject, especially as regards the manners in which they imply different forms of (virtual and symbolic) spatiality. Indeed, the possibilities offered by three-dimensional graphics allow us to recreate entire natural landscapes and indoor environments to explore and interact with [15]. The virtuality of space has been a topic studied by geographers for some time, illustrated by the “imaginative geographies” espoused by Gregory [47]. Actually, video games involve many forms of spatiality. Not only those “internal” to the games themselves (therefore the “virtual” representations of space), but those related to the (“real”) experiences of gamers. Indeed, as is well-known, video gaming is not just a solitary activity: the so-called “multiplayer” mode allows multiple players, through an Internet connection, to participate in the same game, virtually recreating spaces for aggregation and socializing [cf. 15]. In this sense, we can identify two kind of spaces: (i) an “internal” space, virtual yet exploratory, coherent and pertinent within the dynamics of the single game and (ii) an external space of projection, within which multiplayer gamers, in a broader way, interact by challenging/collaborating each other, sharing experiences and knowledges about the games. As Duncan and Gregory state: “all geographies are imaginative geographies—fabrications in the literal sense of ‘something made’—and our access to the world is always made through particular technologies of representation” [48].

The discussion of (virtual, symbolic) space within video games becomes quite complex if we consider the role of AI—an essential component of any video game. In a certain sense, it could be said that AI represents the pattern of behaviour of the opponents and of the game environment: in short, of everything the player (and/or his/her avatar) is confronted with during games.

The believability of a game world always has its weakness in the effectiveness of AI. After all, with all possible efforts, an AI capable of fully confronting and competing with the complexity of human behaviour and choices does not seem within the reach of the current technologies at our disposal. However, it is certain that the continuous evolution of AI, including the heated debate that has spread internationally since 2022 regarding the implementation and use of new AI technologies and platforms (e.g. DALL-E 2, Midjourney, Jasper Chat, ChatGPT), makes the interest of social sciences in the topic ever increasing [49, 50]. From the perspective of cultural geography, studying how AI can change (or, better, is already changing) the way of programming and using the virtual spaces (in and out of games) of video games is a fascinating topic. The case of *free roaming* or *open world* games is an excellent example. The terms indicate a game mode focussed above all on the exploration of a virtual “world” which, especially in recent years, simulates areas with extensions of several square kilometres (up to about 161,600 sq. km). This genre has given rise not only to single games, but to real series composed of several chapters, narratively independent or dependent on the previous episodes. Some of the most famous are undoubtedly series such as “Grand Theft Auto”, “Red Dead Redemption”, “Assassin’s Creed” or “Far Cry”. The episodes of some of these series are released almost annually, meeting a huge success with the public: some of them, for some players, are even the motivation to buy a console or a game PC. Basically, various narratives are set in these miniature worlds, the main one of which is “lived” by the character-protagonist, avatar played by the gamer. Informed by Duncan and Gregory’s insights, as we stated in another study, “video games of this type can be at the same time both imaginary geographies and representation technologies of interactive spatialities” [15]. For instance, as in the case of our earlier study, “Red Dead Redemption 2” has received many awards and is considered by specialized critics as one of the best video games ever produced in recent decades, selling over 35 million copies about two years after its first release (November 2020) [51]. The game, released in 2018, is in its second episode (but, narratively, it ranks as a prequel to the first, released in 2011). Like the first, it allows players to free roam in the western and historical setting of the United States in 1899. As in almost all video games of the same genre, computer graphic sequences alternate with real interactive sessions, where the gamer “impersonates” the protagonist, an outlaw who at the beginning of the game is a member of a gang dedicated to robberies and other criminal activities. During the game the protagonist-gamer will be able to make ethical choices that will have an impact on the general narrative, influencing the interactions (dialogues, fights, alliances, trades, etc.) with the other AI managed characters. Geographically, the game reproduces some north-central and south-central state geographies of North America, with rural and urban landscapes similar to those of late nineteenth century Montana, Wyoming, Dakota, Louisiana and Texas. The reproduced environments,

inspired by the American frontier are quite extensive (about 75 sq. km) and are rendered “credible” both by the great variety of fauna and flora present, and by the effects of light and atmospheric conditions (from sunny to snowy), as well as by the alternation of day/night cycles. The reproduced spaces from the game define very varied environments, rural, urban, and natural landscapes with which the protagonist-gamer can interact in many ways. There is the possibility of collecting plants and hunting wild animals (after studying their behaviour) for the improvement of self-built objects, it is possible (and in many cases it becomes necessary) to use several horses for personal transport (their speed facilitates movement, given the large size of the map), swim in lakes or cross a stream by boat.

The landscapes are “interactive” in a double meaning: both the protagonist-gamer and the flora and fauna are variables that enter in contact modifying each other. Indeed, the narration is articulated around a main story which is structured in such a way as to include various typical North American settings (mountains, plains, canyons, forests, lakes) and then, in secondary plots which favour interaction with the vast playing space. In any case the player, as happens in all open worlds, is free to explore the entire map by various means of transport or on foot and for as long as he wishes. The space, therefore, becomes “agentive” [52] and reproduces various narrative levels, including the one that the player implements outside the main story, through his continuous interaction with the surrounding environment. In this game, as happens in many open worlds, an important role is played by the game’s maps [cf. 15]: the location of the player’s avatar is always displayed on a map in a lower quadrant of the game screen during the gameplay, tracking the journey in a manner similar to Uber or Lyft apps. In addition, the map is used to decide which place to reach or explore. It also reveals new regions and locations as the narrative progresses. As usual, the outermost perimeter of the map is marked in play by invisible walls that the player cannot cross. However, in “Red Dead Redemption 2” it is possible to “cross” the official borders of the map and continue to explore and “counter-map” the *terrae incognitae* that lie beyond by exploiting “glitches”, anomalous behaviours of the software. Some gamers and enthusiasts have also made and uploaded YouTube clips with hundreds of thousands of views [53] in which they explain how to make these “cross-borders”.

Obviously, in free roaming game, the complex reproduced worlds need an AI capable of making the interaction of the player (or his avatar) with the game environment believable. For instance, animals, plants, characters must obey precise physical principles, simulate social behaviours that do not show too many uncertainties, due not only to bugs and programming defects, but above all to an unresponsive AI. Without neglecting the fact that the new potential of AI is also extended to the construction of the same scenarios, some of the creative processes normally managed by graphic, level designers and game designers could be “undermined” by AI, capable of producing good-level graphic works. Therefore, as happens in the case of images and photos of places created artificially by AI, the artistic sector of video game production could also be affected by a poorly controlled use of AI. And this again leads to two attitudes: the enthusiasts of the new possibilities offered by AI and the critics who fear an impoverishment of creativity. Indeed, it is not easy to

take a clear position, above all if one considers the space represented. The landscapes and environments (re)produced in a video game, and above all in those of the free roaming genre, are already systems that present a certain complexity, and which give the player different possibilities of interaction. The aesthetic intention reproducing the different scenarios is actually the very substance of the interaction possibilities, for two reasons. First, some “places” can be more or less attractive and motivate the player to explore them. Second, in many free roaming players have the possibility to take pictures (screenshots) of the environment in which they move and to share them, through official channels, on dedicated websites with other players. This increases and disrupts the “play space” considerably, extending it to the online world through various social media. The environments of a game are certainly regulated, as we said, by a set of rules decided beforehand, but it is true that the evolution of AI could make them increasingly “independent” and “credible” (also from a graphic point of view), e.g. regardless of the “scripts” by their programmers.

The current technological innovation of AI also implements and concerns those of display platforms. Indeed, video games benefit the most from the use of virtual reality (VR): increasingly sophisticated dedicated viewers with 120° field of view with binocular vision make the players’ experience more “immersive”. Obviously, these innovations and technologies find many possible applications. For instance, think of video games for educational as well as recreational purposes which can facilitate the exploration of realistic environments even for students and persons with disabilities [54–56]. Some free roaming games, such as the role-playing “The Elder Scrolls V: Skyrim”, released more than a decade ago are in a new life cycle because they now implement the technology of VR viewers. In this game, the player can interact with a vast but less extensive world than the others mentioned above. However, it is worth remembering that the variety of natural/anthropic environments reproduced is remarkable, especially since it was released in 2011, therefore with a different technology from current games. Moreover, VR technology shows how the spatial dimension is essential for the realization of virtuality: any experience of spatialities implies a corporeality [57], albeit a simulated one.

Another video game, “Flight Simulator”, is an interesting case of this intertwining of space, game and AI. The game, which allows the player to virtually fly using many civilian and some military aircrafts was released in 1982, but the last edition, get out in 2020, uses an innovative technology for the reproduction of space. As emerges from another study [15], in this case the (flyable) map is a kind of 1:1 scale, because the entire surface is reproduced terrestrial in photorealistic quality, including 37,000 airports (commercial, private, airfields, excluding those covered by military secrecy or by restrictions of various kinds) which are practicable today. Cities and the rural areas have been reproduced starting from satellite surveys and proprietary algorithms. Through the technique of the photogrammetry, the game proposes in detail (at least for the top view) over 400 cities. Thanks to the connection to various remote servers, the alternation day/night follows the actual calculated sunrise and sunset times on the coordinates crossed by the virtual aircraft. The software and its AI that manages the simulation, at the discretion of the pilot-player, can acquire in real-time data relating to weather conditions by reproducing them faithfully on video, resulting in a greater

level of realism and “natural” challenges to deal with. Finally, there is another form of video game, interesting from the perspective of the geography of tourism, where players become real tourists motivated by the interest of reaching places that host video game events and activities. These events are becoming more and more popular and include not only fairs but also cultural events [58, 59]. Furthermore, the video game theme is also integrated into events involving live players, such as “live action role playing games” and “alternate reality games”. In “video game” tourism, on the other hand, players become tourists because they have identified a virtual location as a place worthy of being visited and set off in search of a real experience [58, 59].

## 4 Conclusion

This study provides a cultural geographical perspective on how various types of AI can affect some manner in which scholars and users work (or can work) today in geography. The applications are really varied, and it is not possible to foresee the developments unless partially. In fact, the speed at which digital and information technologies improve and make their products complex often makes forecasts obsolete. However, it appears clear that some applications such as those related to some geographical subjects, e.g. the elaboration of GIS, the development of green technologies at the regional level and the geography of risk and disaster studies, can really help scholars to process data and elaborate specific models. On the other hand, in the case of video games the application of AI is even more complex. Video games have become a consolidated and constantly growing form of entertainment in the last thirty years. The development of this industry is closely linked to the broader one of information technology and digital and online platforms. With over three billion gamers worldwide and a market generating over \$184 billion, video games have a significant economic impact and represent a multi-layered cultural product that uses different expressive languages. Here, in particular, we reflected on some issues related to the spatialities of videogames, taking into account some significant cases, such as the “free roaming” and flight simulation. Both let us understand how spaces can be virtually produced and reproduced and how important the role of AI can be in this process. According to Aarseth [60], there is a paradox concerning virtual and real space in/of video games: “Computer games are allegories of space: they pretend to portray space in ever more realistic ways, but rely on their deviation from reality in order to make the illusion playable”. Actually, how we stated, the spatialities of video games include an “internal space” (the represented space in game) and an “external space” that involves players in many ways, such as multiplayer mode to play at a distance, sharing on social networks or on online forums suggestions, explorations of the “worlds”, tricks, videos and so on. In this sense, the space of a video game for players is much wider than one might think. It is a space of interactions and sociality as well as virtual landscapes with which to interact. In this context, AI has and can play a key role. Obviously, it is already an “invisible engine” within video games. From managing the difficulty of the challenges to entertain the players, to

the possibility of extending more and more credible worlds with which to interact, to textual dialogues (chats) with the players, its applications are really many. It is therefore clear that from the perspective of cultural geography, the study of video games, as being “producers of spatiality”, must take into account AIs and how they intervene in the determination of complex game environments. In conclusion, it can be stated that the use and implementation of AI in the various contexts we have talked about here, for geography have epistemological implications that should be the object of further and ongoing studies. Above all, in considering the way in which AIs are developing more and more, we must recognize their diffusion in many areas of our “digital lives”.

**Acknowledgements** The authors thank Prof. Charles Travis for the language review and the referees for the valuable suggestions that helped to improve the quality of the paper.

## References

1. McCulloch W, Pitts W (1990) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biol* 52(1/2):99–115
2. Ferilli S et al (2021) L’Intelligenza Artificiale per lo Sviluppo Sostenibile. CNR, Rome. <https://www.cnr.it/sites/default/files/public/media/attivita/editoria/VOLUME%20FULL%2014%20digital%20LIGHT.pdf>. Accessed 10 Apr 2023
3. Turing AM (1950) Computing machinery and intelligence. *Mind* 49:433–460
4. Alvarez Leon Luis F (2021) AI and the capitalist space economy. *Space Polity* 25(2):220–236. <https://doi.org/10.1080/13562576.2021.1985852>
5. Liu P, Biljecki F (2022) A review of spatially-explicit GeoAI applications in urban geography. *Int J Appl Earth Obs Geoinf* 112:102936
6. Li H, Zech J, Ludwig C, Fendrich S, Shapiro A, Schultz M, Zipf A (2021) Automatic mapping of national surface water with OpenStreetMap and sentinel-2 MSI data using deep learning. *Int J Appl Earth Obs Geoinf* 104
7. Wu AN, Biljecki F (2022) GANmapper: geographical data translation. *Int J Geogr Inf Sci* 36:1394–1422
8. Liu P, Koivisto S, Hiippala T, Van der Lijn C, Väisänen T, Nurmi M, Toivonen T, Vehkakoski K, Pykkinen J, Virmasalo I, Simula M, Hasanan E, Salmikangas A-K, Muukkonen P (2022) Extracting locations from sport and exercise-related social media messages using a neural network-based bilingual toponym recognition model. *J Spatial Inf Sci*
9. Li Y, Yu R, Shahabi C, Liu Y (2018) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: International conference on learning representations
10. Vázquez JJ, Arjona J, Linares M, Casanovas-Garcia J (2020) A comparison of deep learning methods for urban traffic forecasting using floating car data. *Transp Res Procedia* 47:195–202
11. Zhu AX, Lu G, Liu J, Qin CZ, Zhou C (2018) Spatial prediction based on third law of geography. *Ann GIS* 24(4):225–240
12. Li W (2020) GeoAI: where machine learning and big data converge in GIScience. *J Spatial Inf Sci* 20:71–77
13. Janowicz K, Gao S, McKenzie G, Hu Y, Bhaduri B (2020) GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. *Int J Geogr Inf Sci* 34(4):625–636
14. Mai G, Janowicz K, Hu Y, Gao S, Yan B, Zhu R, Cai L, Lao N (2022) A review of location encoding for GeoAI: methods and applications. *Int J Geogr Inf Sci* 36(4):639–673

15. Sabato G (2021) Paesaggi e virtualità: il caso dei videogiochi open world. In: Messina G, D'Agostino L (eds) *Configurazioni e trasfigurazioni. Discorsi sul paesaggio mediato.*, pp 309–320. Nuova Trauben, Torino
16. Gao S (2021) Geospatial artificial intelligence (GeoAI). Oxford Bibliographies. <https://www.oxfordbibliographies.com/display/document/obo-9780199874002/obo-9780199874002-0228.xml>. Accessed 12 Apr 2023
17. VoPham T, Hart JE, Laden F et al (2018) Emerging trends in geospatial artificial intelligence (geoAI): potential applications for environmental epidemiology. *Environ Health* 17:40. <https://doi.org/10.1186/s12940-018-0386-x>
18. Boulos MNK, Peng G, Vopham T (2019) An overview of GeoAI applications in health and healthcare. *Int J Health Geogr* 18. Article No. 7. <https://doi.org/10.1186/s12942-019-0171-2>. <https://ij-healthgeographics.biomedcentral.com/articles>. Accessed 31 Mar 2023
19. Zhu D, Cheng X, Zhang F, Yao X, Gao Y, Liu Y (2020) Spatial interpolation sing conditional generative adversarial neural networks. *Int J Geogr Inf Sci* 34(4):735–758
20. Singleton AD, Longley P (2015) The internal structure of greater London: A comparison of national and regional geodemographic models. *Geo Geogr Environ* 2(1):69–87
21. Gale CG, Singleton AD, Bates AG, Longley PA (2016) Creating the 2011 area lassification for output areas (2011 OAC). *J Spatial Inf Sci* 2016(12):1–27
22. Reades J, De Souza J, Hubbard P (2019) Understanding urban gentrification through achine learning. *Urban Stud* 56(5):922–942
23. Gale CG, Longley PA (2013) Temporal uncertainty in a small area open eodemographic classification. *Trans GIS* 17(4):563–588
24. Torrens PM (2018) Artificial intelligence and behavioral geography. In: Montello DR (ed) *Handbook of behavioral and cognitive geography*. Elgar, Cheltenham, pp 357–371
25. Smith TR (1984) Artificial intelligence and its applicability to geographical problem solving. *Prof Geogr* 36(2):147–158
26. Simon HA (1977) Artificial intelligence systems that understand. In: *Proceedings of the Fifth international joint conference on artificial intelligence*, vol 2. Morgan Kaufmann, Cambridge, MA, pp 1059–1073
27. Newell A, Simon HA (1972) *Human problem solving*. Englewood Cliffs, Prentice-Hall, NJ
28. Cicerone G, Faggian A, Montresor S, Rentocchini F (2023) Regional artificial intelligence and the geography of environmental technologies: does local AI knowledge help regional green-tech specialization? *Region Stud* 57(2):330–343. <https://doi.org/10.1080/00343404.2022.2092610>
29. Li W, Hsu C-Y (2022) GeoAI for large-scale image analysis and machine vision: recent progress of artificial intelligence in geography. *ISPRS Int J Geo-Inf* 11:385. <https://doi.org/10.3390/ijg11107035>
30. Turner A (2006) *Introduction to NeoGeography*, 54 O'Reilly Media, Sebastopol, CA
31. Kuglitsch MM, Pelivan I, Ceola S et al (2022) Facilitating adoption of AI in natural disaster management through collaboration. *Nat Commun* 13:1579. <https://doi.org/10.1038/s41467-022-29285-6>
32. Vinuesa R, Azizpour H, Leite I et al (2020) The role of artificial intelligence in achieving the sustainable development goals. *Nat Commun* 11:233. <https://doi.org/10.1038/s41467-019-14108-y>
33. Newzoo (2022) Global games market report 2022. <https://bit.ly/3KO9ZGn>. Accessed 10 Apr 2023
34. Newzoo (2020) Global games market report 2020. <https://bit.ly/3omLVwh>. Accessed 09 Apr 2023
35. Mukherjee S (2017) *Videogames and postcolonialism: empire plays back*. Palgrave Macmillian, London
36. Hubbell GS (2020) *What is a game? Essays on the nature of videogames*. McFarland & Company, Jefferson, NC
37. Villa M (2020) *Gioco dunque sono. Filosofia del videogamer*. Il nuovo Melangolo, Genova

38. Kriss A (2020) The gaming mind. A new psychology of videogames and the power of play. Little Brown Group, London
39. Herz JC (1997) Joystick Nation: how videogames ate our quarters, won our hearts, and rewired our minds. Little Brown & Co., New York
40. Bittanti M (2004) Per una cultura dei videogames. Teorie e prassi del videogiocare. Unicopli, Milano
41. Valdre M (2007) Dominance and security in the power discourses: Sid Meier's civilization as an example of discourses. In: Power-games, CEU political science journal: the graduate student review, Vol 2, No 1, pp 58–71
42. Nemeslaki A, Molnár L, Nemeslaki TS (2019) Application of computer games in public administration: learning system analysis and technology-society policies with Sid Meyer's civilization. In: Central and Eastern European eDem and eGov days, pp 285–295. <https://doi.org/10.24989/ocg.v335.23>
43. Simons A, Wohlgemant I, Weinmann M, Fleischer S (2020) Good gamers, good managers? A proof-of-concept study with Sid Meier's civilization. RMS. <https://doi.org/10.1007/s11846-020-00378-0>
44. Bogost I (2007) Persuasive games. The expressive power of videogames. The MIT Press, Cambridge (MA)-London
45. Jayemanne D (2017) Performativity in art, literature and videogames. Palgrave Macmillian, London
46. Holmes S (2018) The rhetoric of videogames as embodied practice. Procedural habits. Routledge, New York-London
47. Gregory D (1994) Geographical imaginations. Blackwell, Cambridge MA-Oxford
48. Duncan J, Gregory D (1999) Introduction. In: Duncan J, Gregory D (eds) Writes of passage. Reading travel writing. Routledge, London and New York, pp 1–13
49. Lund BD, Wang T (2023) Chatting about ChatGPT: how may AI and GPT impact academia and libraries? Library Ho Tech News. <https://doi.org/10.1108/LHTN-01-2023-0009>
50. Anders BA (2023) ChatGPT AI in Education. What it is and how to use it in the classroom. Sovorel Publishing Company, USA
51. Game Division (2020) GTA 5 e RDR 2 inarrestabili, le vendite raggiungono cifre da capogiro, 9 November 2020. <https://bit.ly/3fA4Qn>. Accessed 10 Apr 2023
52. Duranti A (2007) Etnopragmatica. La forza nel parlare. Carocci, Roma
53. YouTube: Clips about cross-boundaries (<https://bit.ly/3qgKr8B>; <https://bit.ly/36zoIku>; <https://bit.ly/39Ajwi0>)
54. Cheek DW (2015) A panoramic view of the future of learning and the role of design(ers) in such experiences. In: Hokanson B, Tracey MW, Clinton G (eds) Educational communications and technology: issues and innovations. Springer International Publishing, Switzerland, pp 5–38
55. Kustandi C, Fadhillah DN, Situmorang R, Prawiradilaga DS, Hartati S (2020) VR use in online learning for higher education in Indonesia. Int J Interact Mobile Technol 14(1):31–47
56. Ellis K, Kent M (2011) Disability and new media. Routledge, New York-London
57. Lefebvre H (1990) The production of space. Blackwell, Oxford
58. Dubois L-E, Gibbs C (2018) Video game-induced tourism: a new frontier for destination marketers. Tourism Rev. <https://doi.org/10.1108/TR-07-2017-0115>
59. Sever NS, Sever GN, Kuhzady S (2015) The evaluation of potentials of gamification in tourism marketing communication. Int J Acad Res Bus Soc Sci 5(10):188–202
60. Aarseth E (2001) Allegories of space—the question of spatiality in computer games. In: Eskelinen M, Koskimaa R (eds) Cybertext yearbook 2000. University of Jyväskylä, Publications of the Research Centre for Contemporary Culture, pp 152–171

# S-Transform and Bayesian-Optimized Decision Tree-Based Islanding Detection Approach for Distributed Generation System



Pratyush Kumar Muni, Manohar Mishra, Chinmoy Kumar Patra,  
Debadatta Amresh Gadanayak, and Tanmoy Parida

**Abstract** At present, resources are scarce and difficult to access, leading to an increased focus on renewable resources as a solution. This is where distributed generation (DG) comes into play. However, one of the major challenges in implementing a DG system is islanding, which must be addressed. This report explores the use of Bayesian-Optimized Decision Tree (BODT) for islanding detection in the DG system. The analysis incorporates S-transform techniques, including standard deviation and energy, to distinguish islanding events from non-islanding events, such as short-circuit fault events, load and capacitive switching, and power quality (PQ) events. The goal of this individual's analysis is to improve the health and efficiency of the system by minimizing faults in every way possible. The modeling and simulation were conducted using MATLAB software. The proposed ST-BODT method can be well-suited for real-time islanding detection application as it provides 100% accuracy in without noise situation and 93.9% in 20 dB noisy situation.

**Keywords** Islanding · Bayesian optimization · Decision tree · S-transform

---

P. K. Muni · M. Mishra (✉) · C. K. Patra · D. A. Gadanayak · T. Parida  
Department of Electrical and Electronics Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar 751030, India  
e-mail: [manohar2006mishra@gmail.com](mailto:manohar2006mishra@gmail.com)

D. A. Gadanayak  
e-mail: [debadattagadanayak@soa.ac.in](mailto:debadattagadanayak@soa.ac.in)

T. Parida  
e-mail: [tanmoyparida@soa.ac.in](mailto:tanmoyparida@soa.ac.in)

## 1 Introduction

As individuals' needs continue to grow, the finite nature of resources has resulted in a shortage of available resources. To address this issue, there is a growing interest in switching to renewable resources. However, connecting and distributing power to consumers is a significant challenge for the grid. To overcome this challenge, a controlled and protected system is often selected, such as a microgrid. Microgrids are small-scale electricity distribution networks that can operate independently from the larger power grid by disconnecting from it. They offer several advantages, such as being safe, economical, reliable, and environmentally friendly. However, one of the key issues with microgrids is protection challenges, which play a vital role in the grid's overall performance. To address this issue, islanding detection is necessary to reduce fault conditions and make the system more effective. According to the IEEE standard, the maximum time to detect islanding is 2 s [1]. Early detection is crucial to ensure the safety of line technicians who may not be aware about the events, and they may receive a fatal shock. The primary objective is to make sure that the re-closure remains vulnerable throughout a grid disturbance and provides power to the connected loads. After regaining the supply, the re-closure should be closed to make synchronization with the islanded system and the grid [2].

Islanding is a condition that occurs when a distributed generator continues to supply power to a location even in the absence of external grid electricity. This can create a dangerous situation in the event of a fault in the power system. Islanding can be caused by the excessive utilization of distributed generators such as solar power generators, wind generators, gas turbines, fuel cells, microturbines, within the electrical grid. Therefore, it is important to have a good understanding of distributed generation. Detecting islanding is crucial because it can be fatal if it is not identified in a timely manner. It is essential for the secure and reliable operation of microgrids. Islanding detection is important to prevent substation failure and equipment malfunctioning [3, 4].

The detection of islanding can be classified into two main categories: Remote Detection Method and Local Detection Method [1, 2]. Remote Detection Method is further divided into four schemes: Impedance Insertion Scheme, PLC-based Scheme, PMU-based Scheme, and Transfer Trip Scheme. In the Impedance Insertion Scheme, a small capacitor is inserted into the DG section, which helps to control the flow of reactive power and maintains the stability with respect to the generations and loads. The PLC-based Scheme relies on a low-energy signals that communicated amid the power line communication channels. The PMU-based Scheme detects faults by comparing the synchro-phasor angle and a pre-set threshold angle, while the Transfer Trip Scheme uses a control unit to monitor the circuit breaker. The Local Detection Method is also divided into Active IDMs and Passive IDMs. Passive Islanding Detection Methods (IDMs) are considered advantageous over active IDMs because they do not require any additional equipment or power to detect an islanding condition. Passive IDMs operate by monitoring changes in the system's frequency, voltage, or current, which occur due to the loss of grid connection. Passive IDMs can be more

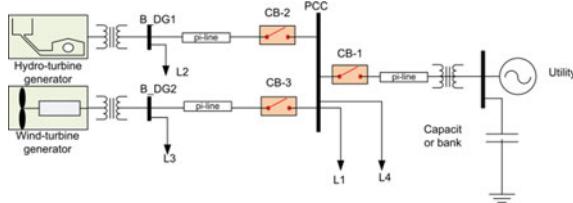
reliable, simpler, and less expensive than active IDMs because they are less prone to false triggering and do not rely on communication between devices [1].

Signal processing techniques are essential in islanding detection methods because they allow for the analysis and identification of changes in the electrical signals generated by the power system. These changes can be used to detect an islanding condition or other power system faults [5]. Signal processing techniques such as the S-transform or wavelet transform can help to identify changes in the frequency or time domain that may be indicative of an islanding event. They are critical in improving the accuracy and effectiveness of islanding detection methods, especially in cases where the power system has a high degree of variability.

This article proposed an islanding detection scheme based on the generalized S-transform. The S-transform is a modern method for analyzing signals that allows for simultaneous examination of the time and frequency domains [6]. In order to categorize various islanding and non-islanding events, Bayesian-Optimized Decision Tree classifier is utilized, which offers improved accuracy and speedy execution compared to other contemporary machine learning models that have been used recently. In this study, several practical events which may mislead the relay and incorrectly detected as islanding cases are considered as non-islanding events such as short-circuit faults (line–ground or line–line–ground or triple lines faults), capacitor bank switching, load switching, and PQ disturbances, which are simulated with varying parameters in order to access its all possibilities. The S-transform is used to calculate different feature indexes, such as energy and standard deviation of negative sequence voltage and current signals. These feature indexes are then used as a feature vector to train BODT for classifying the islanding and non-islanding events. Afterward, the performance of the proposed approach with traditional decision tree approach with similar feature indices is compared and analyzed in details. The performance of the proposed approach is also interpreted under different noisy environments.

## 2 Studied Test System

The research described in this paper investigates a radial distribution network powered by two DGs as shown in Fig. 1. DG-1 is a 10 MW synchronous generator with a hydro-turbine, while DG-2 is a 1.5 MW induction generator with a wind turbine. Both the generators are integrated to a 13 kV–30 km feeder that is linked to a 79 kV grid at the common coupling point. DG-1 operates as the governor system. The system has a base frequency of 60 Hz and a sampling frequency of 2.4 kHz, which equates to 40 samples per cycle. The voltage signal was extracted from the target DG. In Fig. 1, the L1–L4 signifies the distributed loads at the feeder and PCC. Nine different scenarios were examined, including one with islanding. In Case 1, which involves islanding, Table 1 illustrates the state of the circuit breaker in various situations.



**Fig. 1** Studied distribution system [3]

**Table 1** State of the circuit breaker in islanding situations at different DGs

Islanding conditions	Islanding conditions of DGs		Status of Circuit Breaker		
	DG-1	DG-2	CB-1	CB-2	CB-3
First condition	✓	✗	Closed	Open	Closed
Second condition	✗	✓	Closed	Closed	Open
Third condition	✓	✓	Closed	Open	Open

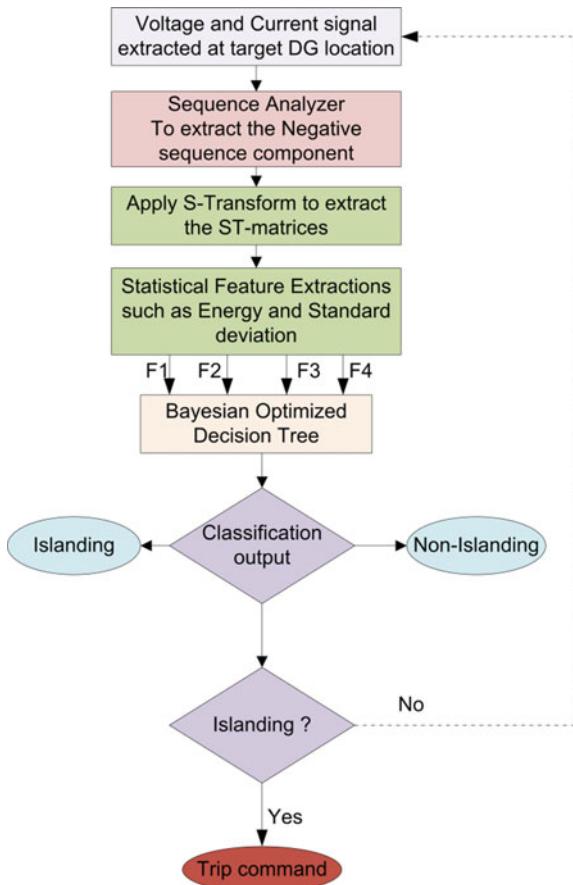
### 3 Methods

Figure 2 shows the flowchart of the proposed ST-BODT approach for detecting loss-of-main conditions in DG system. The process begins with obtaining negative sequence voltage and current signals at the targeted DG's end. In this regard, several islanding and non-islanding signals are generated with all possible system parameter variations. These non-stationary signals were then processed using ST to extract the output time–frequency matrix, which is then used to calculate two statistical features such as the energy and standard deviation. The feature vector comprising these extracted statistical features from both voltage and current signals is then utilized by the Bayesian-Optimized Decision Tree classifier to determine the islanding and non-islanding.

#### 3.1 Feature Extraction Using ST

The ST is a contemporary technique used for signal analysis, which evaluates both the time and frequency domains concurrently [6]. It was developed to serve as an alternative to Fourier-based methods that are not appropriate for analyzing signals that change over time. The ST utilizes a windowed Fourier transform technique, which applies a window function to signals, enabling identification of changes in time-varying frequency characteristics, which is particularly useful for non-stationary signals. In contrast to traditional Fourier-based methods, the ST can adapt the

**Fig. 2** Flowchart of the ST-BODT islanding detection method



frequency resolution according to changes in time, thereby enhancing its effectiveness in non-stationary signal analysis. The ST is an extended version of the wavelet transform that utilizes a Gaussian window that can be scaled and moved to focus on particular regions.

The continuous S-transform can be mathematically expressed as shown in Eq. (1) [3]:

$$S_f(\tau, f) = \int_{-\infty}^{\infty} x(t)g(t, \tau, f)e^{-i2\pi ft} dt, \quad (1)$$

where “ $x(t)$ ” is the signal being analyzed, “ $g(t, \tau, f)$ ” is the Gaussian window function with center time “ $\tau$ ” and center frequency “ $f$ ”, and “ $i$ ” is the imaginary unit.

Similarly, the discrete S-transform can be mathematically expressed as shown in Eq. (2):

$$S_{n,k} = \sum_{m=0}^{N-1} x_m g(m-n, k) e^{-i2\pi km/N}, \quad (2)$$

where “ $x_m$ ” is the discrete signal being analyzed, “ $g(m-n, k)$ ” is the Gaussian window function with center time “ $n$ ” and center frequency “ $k$ ”, “ $N$ ” is the length of the signal, and “ $i$ ” is the imaginary unit.

Both transforms allow for simultaneous analysis of time and frequency characteristics of a signal, with the continuous S-transform being more appropriate for analyzing continuous signals and the discrete S-transform being better suited for analyzing discrete signals. The center time and center frequency of the Gaussian window can be adjusted to localize specific features of interest in the signal. The output of the S-transform is represented as an  $N \times M$  matrix, known as the S-transform amplitude (STA) matrix. The STA matrix contains valuable information about the time-varying frequency characteristics of the analyzed signal, and it can be used to identify and localize specific features of interest in the signal. The STA matrix can also be visualized as a spectrogram, where different colors represent different amplitudes of the signal at different times and frequencies, allowing for easy identification and analysis of signal patterns.

To extract time information from the STA matrix, the maximum absolute value of the matrix is computed using Eq. (3):

$$\text{MAT} = \max(|\text{STA}|), \quad (3)$$

where MAT provides magnitude versus time (in samples) information.

Similarly, to extract frequency information, the maximum absolute value of the transposed STA is computed using Eq. (4):

$$\text{MAF} = \max(|\text{S}^T|), \quad (4)$$

where MAF provides the magnitude versus frequency information.

The negative sequence voltage and current signals extracted at the target DG locations are processed through the ST to generate the STA matrices. Afterward, the following feature sets are derived as shown in Eq. (5).

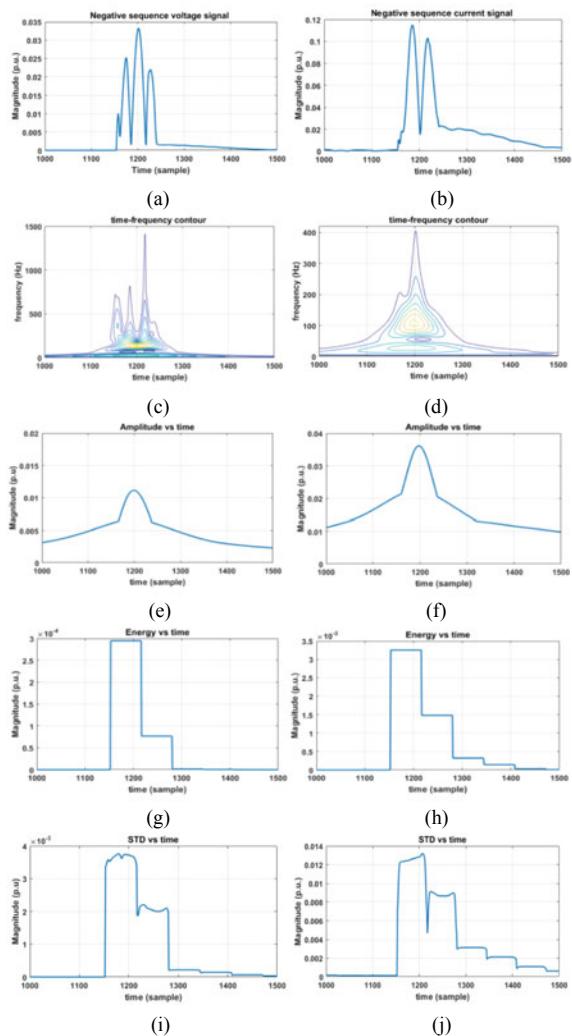
F1 and F2 = maximum value of the energy versus time graph (MET-graph) of the current and voltage signals, respectively.

$$\text{MET} = \max(\text{MAT}^2). \quad (5)$$

F3 and F4 = value of the standard deviation of the MAT plot of the current and voltage signals, respectively.

These features provide important information about the time-varying frequency characteristics of the signal and can be used for various signal processing applications. Figure 3 shows a sample waveform related to ST application during islanding events at DG-1 ends.

**Fig. 3** ST applied to islanded signal extracted at target DG-1 with 10% power mismatch



### 3.2 Bayesian Optimization (BO)

BO is a powerful machine learning technique that is employed to optimize complex functions that are expensive to assess [7, 8]. This method is particularly useful when the objective function is unknown, noisy, or computationally expensive to compute. Bayesian optimization utilizes a probabilistic model to approximate the objective function and optimize it. The algorithm begins by selecting a set of initial hyperparameters, which are then used to evaluate the objective function. The results of this evaluation are then used to update the probabilistic model and create a posterior distribution over the objective function. The posterior distribution is then used to choose

the next set of hyperparameters to evaluate. This process is repeated until an optimal set of hyperparameters is found that optimizes the objective function [7]. Bayesian optimization has proven to be highly effective in various applications, such as fine-tuning machine learning models by optimizing their hyperparameters, optimizing computationally intensive simulations, and controlling robotics. This technique has shown to be an efficient and effective approach to solve complex optimization problems, especially in cases where the objective function is difficult to evaluate or where the evaluation of the function is computationally expensive. The primary objective of the Bayesian optimization algorithm is to identify the value of  $x$  that minimizes the scalar objective function  $g(x)$ . The output of the algorithm can vary depending on whether the function is deterministic or stochastic. In order to minimize the objective function  $g(x)$ , the Bayesian optimization algorithm employs three key components: a Gaussian process model that represents  $g(x)$ , a Bayesian update process that modifies the Gaussian model after each evaluation of  $g(x)$ , and an acquisition function,  $aq(x)$ , which is optimized to determine the next evaluation point. The acquisition function is designed to identify potential increases in  $g(x)$ , while avoiding values that could inflate it. Therefore, the optimal decision is made by calculating the expected improvement ( $E_i$ ) using the following formula:

$$E_i(x, m) = E_m[\max(0, \mu_m(x_{\text{best}})g(x))]. \quad (6)$$

Here,  $x_{\text{best}}$  refers to the position where the posterior mean is at its minimum, and  $\mu_m(x_{\text{best}})$  represents the lowest value of the posterior mean.

### 3.3 BO Decision Tree

The decision tree is a widely used machine learning algorithm that addresses classification and regression problems by creating a tree-like structure [9]. The tree is composed of internal nodes that represent features or attributes, branches that indicate decision rules based on these features, and leaves that denote the resulting classifications or regression values. The algorithm works by recursively splitting the data into smaller subsets based on the most informative attributes until a stopping criterion is met. The stopping criteria may include limiting the depth of the tree, setting a minimum number of samples required for splitting nodes, or specifying a minimum threshold for the improvement of the selected evaluation metric. One of the significant advantages of decision trees is their interpretability, which means that the decision-making process is transparent and understandable. Moreover, they can handle a combination of numerical and categorical data, missing values, and outliers, making them versatile in various applications. However, if not appropriately regularized, decision trees can overfit the training data and may not perform as well as other machine learning algorithms in predicting new data points.

In this work, the usual decision tree is modified through integrating the Bayesian optimization algorithm in order to tune the hyperparameters during training stage. In

this regard, initially the islanding and non-islanding data were arranged into original datasets with a matrix size  $M \times N$  and a target vector indicating the class labels of each sample, where  $M$  is the number of samples and  $N$  is the number of features. This dataset is then sub-divided into training and testing sets. Afterward, a “ClassificationTree” object is created in MATLAB to represent the decision tree classifier. In the next stage, a Bayesian optimization algorithm is used to find the optimal values for the hyperparameters such as minimum number of samples per leaf node and the number of splits per node. Once the optimal hyperparameters have been found, decision tree classifier is trained using the training set and the optimized hyperparameters. To do this, “MaxNumSplits”, “MinLeafSize”, and “NumVariablesToSample” properties of the “ClassificationTree” object are set corresponding to the obtained optimized values and called the “fit” method of the object, followed by passing the training set and target vector as arguments. Output-trained Bayesian-Optimized Decision Tree classifier is then used for the classification tasks for the testing dataset.

## 4 Islanding Assessment Results and Discussion

In order to test the proposed approach, a total of 600 islanding events and 800 non-islanding events were simulated. The islanding signals are simulated by introducing active and reactive power mismatch of up to 50%. Similarly, the non-islanding signal is generated by varying the line and system parameters such as fault resistance varying from 0 to  $100 \Omega$  for short-circuit fault case, large amount of loads switching (ranging 5–30 MW), voltage amplitude variation of 10–80% for sag and swell case, and large value of capacitor bank (ranging 0.5–10 MVar) switching. In the followed subsections, the performance of the BODT classifier using ST-based feature vectors is presented. Here, (60:40)% of data from both islanding and non-islanding cases are used for training and testing, respectively. The cumulative training data for the proposed classifier includes pure signal (without noise) as well as noised data having  $\text{SNR} = 40 \text{ dB}$  and  $20 \text{ dB}$ . For the evaluation of the classifier, a fivefold cross-validation (FCV) is carried out on the training data.

### 4.1 Performance of the BODT During Training Phase (5-FCV)

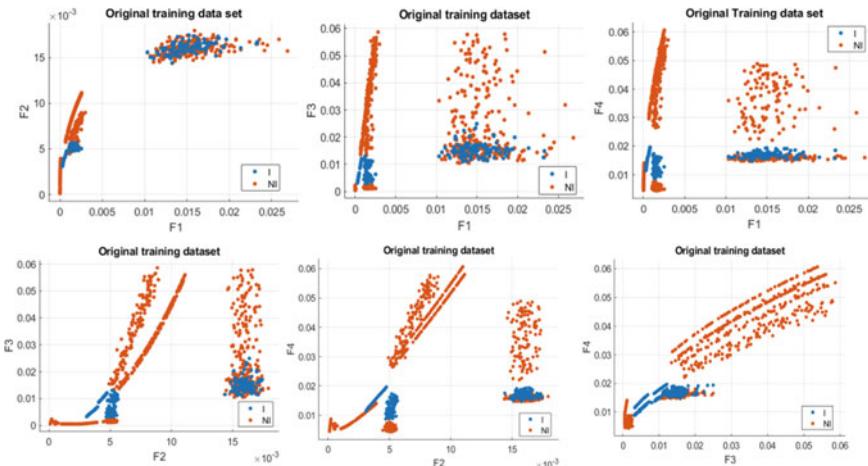
It is well-known that the power signal extracted in the real time may come with several distortions and noises. Therefore, in this study, we have formulated the training dataset with respect to both pure and different values of noise signals. In this regard, the total training data comprise 3542 samples (1080 islanding and 2462 non-islanding). As the training data comprise different noise values, it is important to analyze the scatter plots between different attributes. The scatter plots between all

four features are presented in Fig. 3. Here, the complex relationships between the attributes can be easily analyzed.

The performance results related to the 5-FCV process in the training phase are pictorially represented in Figs. 4, 5, and 6. Figure 4 shows a sample result related to the optimization of the hyperparameter of DT utilizing the Bayesian optimization algorithm. The curves related to the minimum error rate with respect to the iterations are plotted in Fig. 5. In this curve, the red dot shows the best point hyperparameter value.

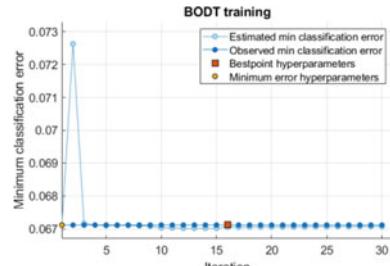
Figure 6a provides the results in terms of confusion matrix related to the measurement at target DG-1, and Fig. 6b provides the results related to the measurement at target DG-2. It can be seen that for both DG ends, the BODT performs better compared to the traditional DT in terms of true-positive rate and false-negative rate. The rate of convergence and area under the curve related to the results of Fig. 6 are presented in Fig. 7, respectively.

The validation performance of the proposed ST-BODT method as well as traditional ST-DT model to classify the islanding and non-islanding cases is calculated

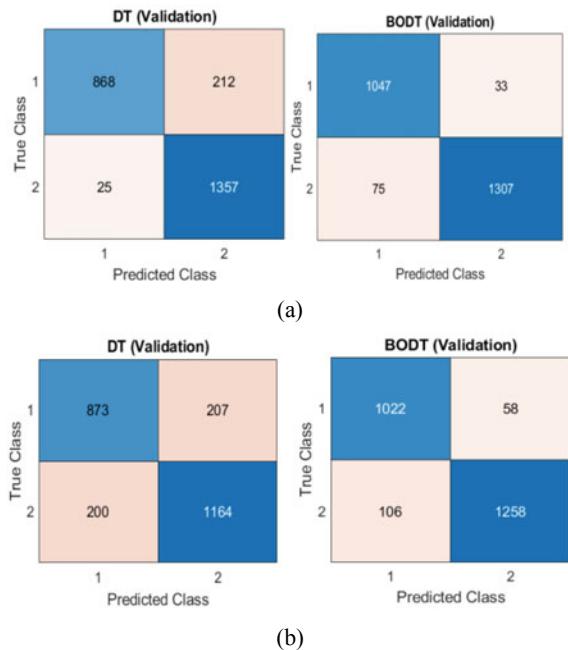


**Fig. 4** Scatter plot between features

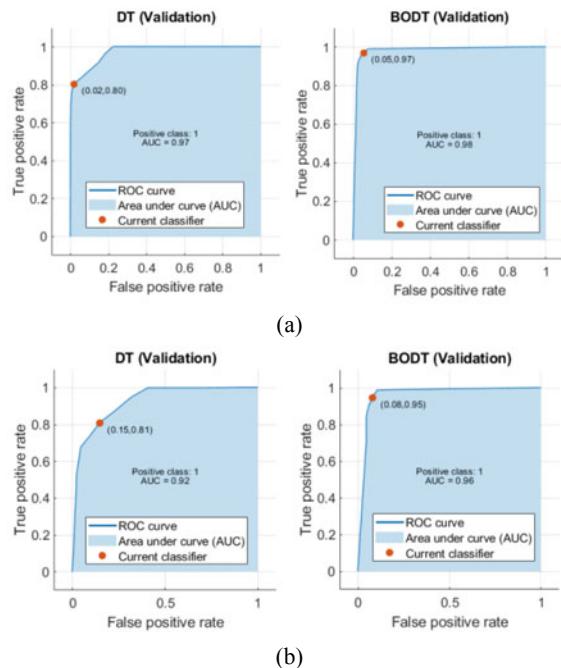
**Fig. 5** BODT training process for DG-2-based dataset (iteration versus minimum classification error)



**Fig. 6** Confusion matrix related to the fivefold cross-validation: **a** at target DG-1, **b** at target DG-2



**Fig. 7** ROC curve and area under the curve during fivefold cross-validation **a** at target DG-1, **b** at target DG-2



**Table 2** Performance parameter during training with fivefold cross-validation

Model	Target DG	Dependability (%)	Security (%)	Accuracy (%)
DT	DG-1	80.4	98.2	90.4
	DG-2	80.8	85.3	83.3
BODT	DG-1	96.9	94.6	95.6
	DG-2	94.6	92.2	93.3

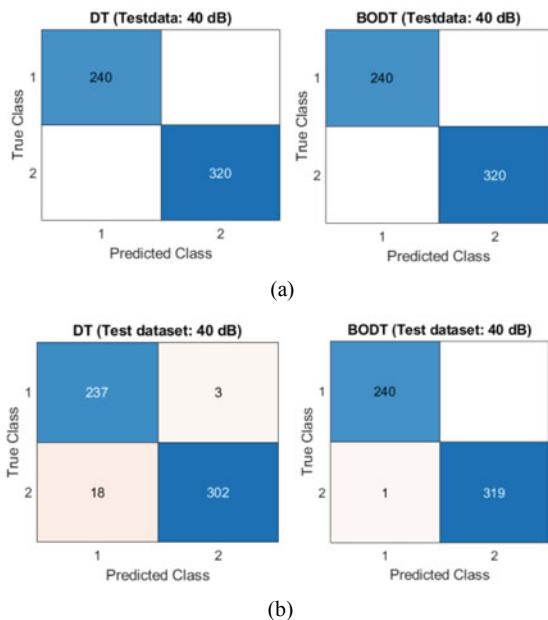
from the confusion matrix and tabulated in Table 2. The performance accuracy of ST-BODT model is found to be 95.6 and 93.3% for DG-1 and DG-2 cases, respectively. It can be seen that the ST-BODT model outperforms to ST-DT model with margin of 5–10%.

#### 4.2 *Performance of the BODT During Testing Phase*

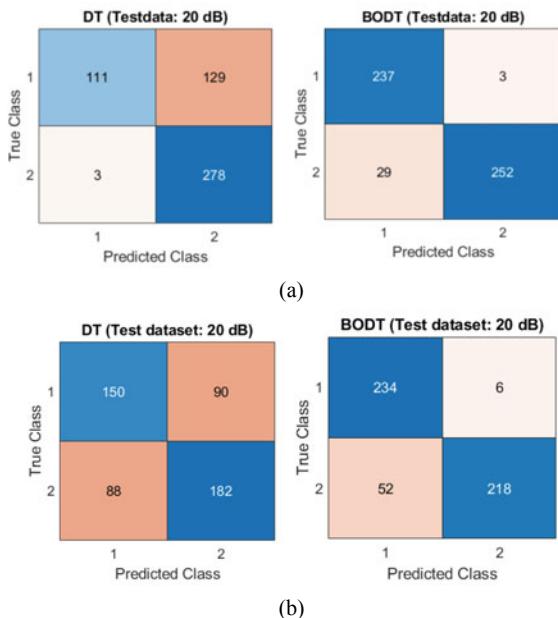
After the training process is validated, the trained model is saved for the future use as a black box solution in the data mining task. In this work, the 40 and 20 dB noise-associated testing data (i.e., 40% of total dataset) are used for testing process individually on the output trained model. The results with respect to the 40 dB noisy data and 20 dB noisy data are pictorially presented in Figs. 8 and 9, respectively. It has been seen that in 40 dB noisy case, incorrect classification rate is zero or very least value for DG-1 end or DG-2 end measuring samples, whereas these values are very highly increased in the case of 20 dB noisy dataset. The rate of convergence and area under the curve related to the result of Fig. 9, are presented in Fig. 10 respectively.

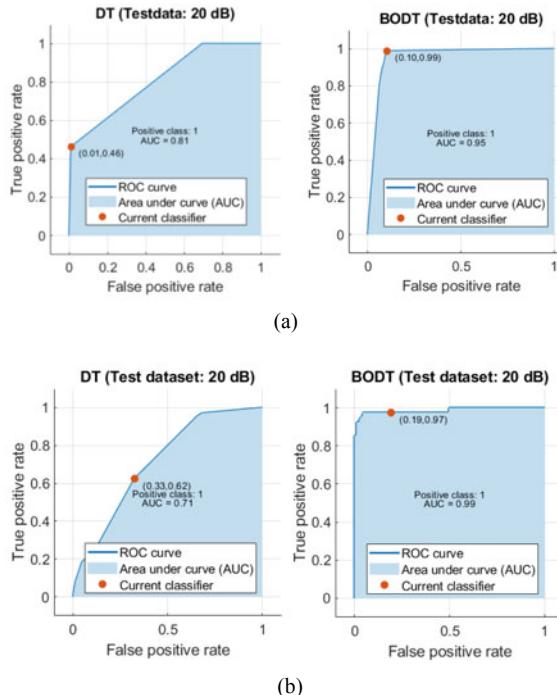
The testing performance of the proposed ST-BODT method as well as traditional ST-DT model to classify the islanding and non-islanding cases is calculated from the confusion matrix as shown in Figs. 9 and 10 and tabulated in Table 3. The performance accuracy of ST-BODT model is found to be 93.9% and 88.6% for DG-1 and DG-2 cases, respectively, during accessing the 20 dB dataset. It can be seen that the ST-BODT model outperforms to ST-DT model with margin of 15–20% in 20 dB noisy case.

**Fig. 8** Confusion matrix related to the 40 dB noisy test data: **a** at target DG-1, **b** at target DG-2



**Fig. 9** Confusion matrix related to the 20 dB noisy test data: **a** at target DG-1, **b** at target DG-2





**Fig. 10** ROC curve and area under the curve to the 20 dB noisy test data: **a** at target DG-1, **b** at target DG-2

**Table 3** Performance parameter with testing dataset

Model	Target DG	Dependability (%)	Security (%)	Accuracy (%)
<i>40 dB noisy case</i>				
DT	DG-1	100	100	100
	DG-2	98.8	94.4	96.2
BODT	DG-1	100	100	100
	DG-2	100	99.7	99.8
<i>20 dB noisy case</i>				
DT	DG-1	46.2	98.9	74
	DG-2	62.5	67.4	65
BODT	DG-1	98.8	89.7	93.9
	DG-2	97.5	80.7	88.6

## 5 Conclusion

This paper presents an improved decision tree-based method for islanding detection and classification utilizing the ST-based features. The accuracy of the traditional decision tree is improved through optimizing its hyperparameters such as minimum number of samples per leaf node and the number of splits per node. In this regard, a Bayesian optimization algorithm is utilized. Performance of the proposed ST-BODT method is analyzed in both training and testing phases. In the training phase, the fivefold cross-validation is used to validate the training model. Afterward, the trained model is tested with different test datasets. It can be observed that the proposed approach provides an excellent result compared to traditional decision tree model. As the decision tree classifier is well-known for its speed of detection and ST is advantageous owing to its noise immunity capability, the proposed ST-BODT method is well-suited for real-time islanding detection application. It is concluded that the proposed method is very simple, straightforward, and easy to implement with minimal computational time compared to other time-frequency-based techniques.

## References

1. Mishra M, Chandak S, Rout PK (2019) Taxonomy of islanding detection techniques for distributed generation in microgrid. *Renew Energy Focus* 31:9–30
2. Cebollero JA, Cañete D, Martín-Arroyo S, García-Gracia M, Leite H (2022) A survey of islanding detection methods for microgrids and assessment of non-detection zones in comparison with grid codes. *Energies* 15(2):460
3. Mishra M, Rout PK, Sahu R, Ray D, Swarup S (2016) Study the performance of S-transform based extreme learning machine for islanding detection in distributed generation. In: 2016 National power systems conference (NPSC). IEEE, pp 1–6
4. Sharma R, Singh P (2012) Islanding detection and control in grid based system using wavelet transform. In 2012 IEEE fifth power India conference. IEEE, pp 1–4
5. Mishra M, Patnaik B, Bansal RC, Naidoo R, Naik B, Nayak J (2021) DTCDWT-SMOTE-XGBoost-based islanding detection for distributed generation systems: an approach of class-imbalanced issue. *IEEE Syst J* 16(2):2008–2019
6. Mishra M, Rout PK (2019) Fast discrete s-transform and extreme learning machine based approach to islanding detection in grid-connected distributed generation. *Energy Syst* 10:757–789
7. Asante-Okyere S, Shen C, Osei H (2022) Enhanced machine learning tree classifiers for lithology identification using Bayesian optimization. *Appl Comput Geosci* 16:100100
8. Injadat M, Salo F, Nassif AB, Essex A, Shami A (2018) Bayesian optimization with machine learning algorithms towards anomaly detection. In: 2018 IEEE global communications conference (GLOBECOM). IEEE, pp. 1–6
9. Pragati A, Gadanayak DA, Parida T, Mishra M (2023) Decision tree based multi-terminal VSC-HVDC transmission line protection scheme. In: 2023 International conference for advancement in technology (ICONAT). IEEE, pp 1–4