

R. Jagadeesh Kannan
Sabu M. Thampi
Shyh-Hau Wang *Editors*

Computer Vision and Machine Intelligence Paradigms for SDGs

Select Proceedings of ICRTAC- CVMIP
2021

Lecture Notes in Electrical Engineering

Volume 967

Series Editors

Leopoldo Angrisani, Department of Electrical and Information Technologies Engineering, University of Napoli Federico II, Naples, Italy

Marco Arteaga, Departament de Control y Robótica, Universidad Nacional Autónoma de México, Coyoacán, Mexico

Bijaya Ketan Panigrahi, Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, Delhi, India

Samarjit Chakraborty, Fakultät für Elektrotechnik und Informationstechnik, TU München, Munich, Germany

Jiming Chen, Zhejiang University, Hangzhou, Zhejiang, China

Shanben Chen, Materials Science and Engineering, Shanghai Jiao Tong University, Shanghai, China

Tan Kay Chen, Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore

Rüdiger Dillmann, Humanoids and Intelligent Systems Laboratory, Karlsruhe Institute for Technology, Karlsruhe, Germany

Haibin Duan, Beijing University of Aeronautics and Astronautics, Beijing, China

Gianluigi Ferrari, Università di Parma, Parma, Italy

Manuel Ferre, Centre for Automation and Robotics CAR (UPM-CSIC), Universidad Politécnica de Madrid, Madrid, Spain

Sandra Hirche, Department of Electrical Engineering and Information Science, Technische Universität München, Munich, Germany

Faryar Jabbari, Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA, USA

Limin Jia, State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

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

Alaa Khamis, German University in Egypt El Tagamoa El Khames, New Cairo City, Egypt

Torsten Kroeger, Stanford University, Stanford, CA, USA

Yong Li, Hunan University, Changsha, Hunan, China

Qilian Liang, Department of Electrical Engineering, University of Texas at Arlington, Arlington, TX, USA

Ferran Martín, Departament d'Enginyeria Electrònica, Universitat Autònoma de Barcelona, Bellaterra, Barcelona, Spain

Tan Cher Ming, College of Engineering, Nanyang Technological University, Singapore, Singapore

Wolfgang Minker, Institute of Information Technology, University of Ulm, Ulm, Germany

Pradeep Misra, Department of Electrical Engineering, Wright State University, Dayton, OH, USA

Sebastian Möller, Quality and Usability Laboratory, TU Berlin, Berlin, Germany

Subhas Mukhopadhyay, School of Engineering and Advanced Technology, Massey University, Palmerston North, Manawatu-Wanganui, New Zealand

Cun-Zheng Ning, Electrical Engineering, Arizona State University, Tempe, AZ, USA

Toyoaki Nishida, Graduate School of Informatics, Kyoto University, Kyoto, Japan

Luca Oneto, Department of Informatics, BioEngineering, Robotics and Systems Engineering, University of Genova, Genova, Genova, Italy

Federica Pascucci, Dipartimento di Ingegneria, Università degli Studi "Roma Tre", Rome, Italy

Yong Qin, State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

Gan Woon Seng, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Joachim Speidel, Institute of Telecommunications, Universität Stuttgart, Stuttgart, Germany

Germano Veiga, Campus da FEUP, INESC Porto, Porto, Portugal

Haitao Wu, Academy of Opto-electronics, Chinese Academy of Sciences, Beijing, China

Walter Zamboni, DIEM—Università degli studi di Salerno, Fisciano, Salerno, Italy

Junjie James Zhang, Charlotte, NC, USA

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

R. Jagadeesh Kannan · Sabu M. Thampi ·
Shyh-Hau Wang
Editors

Computer Vision and Machine Intelligence Paradigms for SDGs

Select Proceedings of ICRTAC-CVMIP 2021

 Springer

Editors

R. Jagadeesh Kannan
School of Computer Science
and Engineering
Vellore Institute of Technology Chennai
Chennai, Tamil Nadu, India

Sabu M. Thampi
School of Computer Science
and Engineering (SoCSE)
Kerala University of Digital Sciences
Trivandrum, Kerala, India

Shyh-Hau Wang
Department of Computer Science
and Information Engineering
National Cheng Kung University
Tainan, Taiwan

ISSN 1876-1100

ISSN 1876-1119 (electronic)

Lecture Notes in Electrical Engineering

ISBN 978-981-19-7168-6

ISBN 978-981-19-7169-3 (eBook)

<https://doi.org/10.1007/978-981-19-7169-3>

© 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

Preface

Sustainable Development Goals stated by United Nations projects a call for the entire world to start acting towards 17 goals such as providing an end towards poverty and protecting the planet to ensure peace and prosperity by 2030. Taking advantage of digitization to provide part of solutions to achieve these goals, the conference ICRTAC'21 particularly aims to project and provide computer vision and machine intelligence-based techniques for achieving peace. With the entire world moving towards Industry 4.0, it is becoming an essential context for the research community to suggest ICT-based techniques for realizing the UN's goals. Computer Vision and Machine Learning approach play a vital role in current research trends like the Internet of Things, brain human interfaces, etc. Today, the large data termed big data that is required for injecting intelligence is available easily and is growing exponentially. Researchers can take advantage of this big data, trained with intelligent vision techniques to interpret, understand and reflect the visual world in a better and easier way. The contents of the book will encourage all the research minds to nurture the current technological needs and to provide a broad view of the new evolving techniques to address the various societal issues prevailing in the sustainable development field. This book discusses the technical content for connecting like-minded people and industry peers to share their ideas, find solutions to their problems and work with industry peers. This book provides updates on key issues and highlights recent advances in a single broad topic applicable to many different sub-fields by exploring various multidisciplinary technologies. For instance, computer vision combined with machine intelligence is spreading its wings in various domains like mobile applications, business tools, IoT, lifestyle, health, social media, finance, automobile, robotics, etc. This book supports the transfer of vital knowledge to the next generation of researchers, students, and academicians. The major intent of the publication is:

- To explore the potential usage of smart techniques to provide innovative solutions in image analysis.
- To investigate novel and state-of-the-art methods in computer vision coupled with intelligent techniques including machine learning, deep learning, and soft computing techniques.
- To provide a prominent interdisciplinary platform for researchers, practitioners, and educators to deliver and discuss the contemporary innovations, trends, and concerns in computer vision thereby suggesting solutions to real-world problems adhering to sustainable development.

Chennai, India
Trivandrum, India
Tainan, Taiwan

R. Jagadeesh Kannan
Sabu M. Thampi
Shyh-Hau Wang

Contents

PTZ-Camera-Based Facial Expression Analysis using Faster R-CNN for Student Engagement Recognition	1
E. Komagal and B. Yogameena	
Convergence Perceptual Model for Computing Time Series Data on Fog Environment	15
Rupa Kesavan, S. Poorani, R. Iyswarya, S. U. Muthunagai, R. Anitha, and L. Vijayaraja	
Localized Super Resolution for Foreground Images Using U-Net and MR-CNN	25
Umashankar Kumaravelan and M. Nivedita	
SMS Spam Classification Using PSO-C4.5	41
D. Saraswathi and D. Sowmya	
Automated Sorting, Grading of Fruits Based on Internal and External Quality Assessment Using HSI, Deep CNN	49
P. Rahul Ganesh, R. Priyatharshini, M. Sarath Kumar, and A. Raj Kumar	
Pest Detection Using Improved YOLO Architecture	59
M. Sujaritha, M. Kavitha, and S. Roobini	
Classification of Fungi Effected Psidium Guajava Leaves Using ML and DL Techniques	69
Sukanya S. Gaikwad, Shivanand S. Rumma, and Mallikarjun Hangarge	
Deep Learning Based Recognition of Plant Diseases	83
Swetha Parthiban, Sneha Moorthy, Sribalaji Sabanayagam, Shobana Shanmugasundaram, Athishwaran Naganathan, Mohan Annamalai, and Sabitha Balasubramanian	
Artificial Cognition of Temporal Events Using Recurrent Point Process Networks	95
N. Bala Sundara Ganapathy and M. Deeptavarna	

Performance Analysis of Energy Efficient Video Transmission Using LEACH Based Protocol in WSN	103
M. Radhika and P. Sivakumar	
Hybridization of Texture Features for Identification of Bi-Lingual Scripts from Camera Images at Wordlevel	113
Satishkumar Mallappa, B. V. Dhandra, and Gururaj Mukarambi	
Advanced Algorithmic Techniques for Topic Prediction and Recommendation—An Analysis	125
Shaik Nazeer, Prathyusha Yayavaram, and L. Jani Anbarasi	
Implementation of an Automatic EEG Feature Extraction with Gated Recurrent Neural Network for Emotion Recognition	133
Rajeswari Rajesh Immanuel and S. K. B. Sangeetha	
High Performance Classifier for Brain Tumor Detection Using Capsule Neural Network	151
J. S. Thanga Purni, R. Vedhapriyavadhana, S. L. Jayalakshmi, and R. Girija	
Mining Suitable Symptoms to Identify Disease Using Apriori and NBC	165
R. Sandha and M. Vasumathy	
Background Features-Based Novel Visual Ego-Motion Estimation	175
B. Sharmila and D. Nedumaran	
Livspecs: Design and Implementation of Smart Specs for Hearing and Visually Challenged Persons	191
P. K. Prithvi, K. Chandru, Krishnan B. Yashwanth, Fathima M. Shabika, R. Ranjana, and T. Subha	
Self-balancing Robot Using Arduino and PID Controller	201
C. N. Savithri, R. S. Roopesh, N Lavanya Devi, P. Shanthakumar, and P. Thirumurugan	
A Survey Based on Online Voting System Using Blockchain Technology	209
S. P. Shanthinii, M. Usha, and P. Prittopaul	
Survey on Collaborative Filtering Technique for Recommender System Using Deep Learning	217
S. L. Jothilakshmi and R. Bharathi	
A Survey on Power Consumption Indicator Using Machine Learning-Based Approach	227
R. Hamsini and P. Visu	

A Novel Hand Gesture Recognition for Aphonic People Using Convolutional Neural Network 235
S. Deepa, A. Umamageswari, and S. Menaka

Comprehensive Analysis of Defect Detection Through Image Processing and Machine Learning for Photovoltaic Panels 245
S. Prabhakaran, R. Annie Uthra, and J. Preetharoselyn

Covid Analysis Prediction Using Densenet Method in Deep Learning 263
M. Usha, P. Prittopaul, and D. Lekha

Feature Extraction Based on GLCM and GLRM Methods on COVID-19 Dataset 271
N. Suganthi and K. Sarojini

Memory Augmented Distributed Monte Carlo Tree Search Algorithm-Based Content Popularity Aware Content Recommendation Using Content Centric Networks 281
A. P. Christopher Arokiaraj and D. Hari Prasad

Enhancing Protection Against Scalper Bots with ML 293
R. Gowthamani, K. Sasi Kala Rani, C. R. Jayanth, B. Jeba Regan Raj, and J. Binesh

An Enhanced Optimized Abstractive Text Summarization Traditional Approach Employing Multi-layered Attentional Stacked LSTM with the Attention RNN 303
M. Nafees Muneera and P. Sriramy

Implementation of Machine Learning Methods on Data to Analyze Emotional Health 319
Vijaya Chandra Jadala, Sai Kiran Pasupuleti, S. Hrushikesava Raju, Srihari Babu Gole, N. Ravinder, and B. Sreedhar

Course Difficulty Estimation Based on Mapping of Bloom’s Taxonomy and ABET Criteria 329
M. Premalatha, G. Suganya, V. Viswanathan, and G. Jignesh Chowdary

Editors and Contributors

About the Editors

R. Jagadeesh Kannan is a Professor in the School of Computer Science and Engineering at Vellore Institute of Technology, India. He completed his Ph.D. in Handwritten Character Recognition using Hybrid Techniques from Anna University, Chennai, India. He got his M.E. degree in Computer Science and Engineering from National Engineering College, Tamil Nadu, and B.E. in Instrumentation and Control Engineering from Madurai Kamaraj University, Tamil Nadu, India. Professor Kannan has over 18 years of teaching and industrial experience in reputed organizations. Professor Kannan has got several publications in conference proceedings and journals of national and international repute. His research interests are neural networks, fuzzy logic, neuro-fuzzy systems, soft computing tools, pattern recognition, natural language processing, image processing, networking, printed, handwritten and cursive character recognition, and artificial intelligence.

Sabu M. Thampi is a Professor at the School of Computer Science and Engineering (SoCSE), Kerala University of Digital Sciences, Innovation and Technology (KUDSIT)—formerly IIITM-K, Technocity Campus, Trivandrum, Kerala, India. He holds a Ph.D. in Computer Engineering from the National Institute of Technology Karnataka. His current research interests include the Internet of things (IoT), cognitive security, social networks, endpoint security, and smart cyber-physical systems. Sabu is also coordinating the Connected Systems and Intelligence (CSI) Lab at KUDSIT. Dr. Sabu has been actively involved in funded research projects and published papers in book chapters, journals, and conference proceedings. He has authored and edited a few books, as well as edited 45+ conference proceedings published by Springer in various series, as well as a few others published by IEEE, ACM, and Elsevier. Sabu has served as Guest Editor for special issues in a few journals and a program committee member for many international conferences and workshops. He has co-chaired several international workshops and conferences. He has initiated (ISTA, SSCC, SIRS, ISI, SoMMA, ACN, and CoCoNet...) and is also

involved in the organization of several annual conferences/symposiums. Sabu is currently serving as Editor for Elsevier Journal of Network and Computer Applications (JNCA), Elsevier Journal of Engineering Applications of Artificial Intelligence (2022), Connection Science—Taylor & Francis; Associate Editor for IEEE Access and International Journal of Embedded Systems, Inderscience, UK; and Reviewer for several reputed international journals. Sabu is a Senior Member of IEEE and ACM. He is the Founding Chair of the ACM Trivandrum Professional Chapter.

Shyh-Hau Wang is a Professor in the Department of Computer Science and Information Engineering (CSIE); Institute of Medical Informatics (IMI); MS Program in Artificial Intelligence Technology at National Cheng Kung University (NCKU), Tainan, Taiwan. He completed his Ph.D. in Bioengineering at the Pennsylvania State University (PSU), USA. His research interests include medical and health-care informatics, ultrasound signal and imaging system, machine/deep learning for signal/image processing, and smart ultrasound tissue characterization. He is a Senior Member of IEEE Engineering in Medicine and Biology Society (EMBS) and IEEE Ultrasonics, Ferroelectrics, and Frequency Control (UFFC). He has many publications to his credit and served as editor for books in the areas of his research interest.

Contributors

L. Jani Anbarasi School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

R. Anitha Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India

Mohan Annamalai Kumaraguru College of Technology, Coimbatore, India

R. Annie Uthra SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

N. Bala Sundara Ganapathy Panimalar Engineering College, Chennai, India

Sabitha Balasubramanian Kumaraguru College of Technology, Coimbatore, India

R. Bharathi University College of Engineering, Nagercoil, Tamil Nadu, India

J. Binesh Sri Krishna College of Engineering and Technology, Coimbatore, India

K. Chandru Sri Sairam Engineering College, Chennai, India

A. P. Christopher Arokiaraj Department of Computer Applications, Sri Ramakrishna College of Arts and Science, Coimbatore, India

S. Deepa Department of CSE, SRM Institute of Science and Technology, Chennai, India

M. Deeptavarna Panimalar Engineering College, Chennai, India

B. V. Dhandra Department of Statistics, Christ (Deemed to Be University), Bengaluru, India

Sukanya S. Gaikwad Department of Computer Science, Gulbarga University, Kalaburagi, Karnataka, India

R. Girija CST Department, Manav Rachna University, Faridabad, India

Srihari Babu Gole Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India

R. Gowthamani Sri Krishna College of Engineering and Technology, Coimbatore, India

R. Hamsini Velammal Engineering College, Chennai, India

Mallikarjun Hangarge Department of Computer Science, Karnatak Arts, Science, and Commerce College, Bidar, Karnataka, India

D. Hari Prasad Department of Computer Applications, Sri Ramakrishna College of Arts and Science, Coimbatore, India

S. Hrushikesava Raju Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India

Rajeswari Rajesh Immanuel Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India

R. Iyswarya Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India

Vijaya Chandra Jadala Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India

S. L. Jayalakshmi Department of Computer Science (Main Campus), School of Engineering and Technology, Pondicherry University, Puducherry, India

C. R. Jayanth Sri Krishna College of Engineering and Technology, Coimbatore, India

B. Jeba Regan Raj Sri Krishna College of Engineering and Technology, Coimbatore, India

G. Jignesh Chowdary Vellore Institute of Technology, Chennai, India

S. L. Jothilakshmi Amrita College of Engineering and Technology, Nagercoil, Tamil Nadu, India

M. Kavitha Sri Krishna College of Engineering and Technology, Coimbatore, India

Rupa Kesavan Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India

E. Komagal Department of Electronics and Communication Engineering, Latha Mathavan Engineering College, Madurai, Tamil Nadu, India

Umashankar Kumaravelan Ydealogy Ventures, Chennai, India

N Lavanya Devi Department of Electronics and Communication Engineering, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India

D. Lekha Department of CSE, RMK College of Engineering and Technology, Thiruvallur, Tamil Nadu, India

Satishkumar Mallappa Department of P.G.Studies and Research in Computer Science, Gulbarga University, Gulbarga, India

S. Menaka Department of CSE, SRM Institute of Science and Technology, Chennai, India

Sneha Moorthy Kumaraguru College of Technology, Coimbatore, India

Gururaj Mukarambi School of Computer Science, Department of Computer Science, Central University of Karnataka, Gulbarga, India

S. U. Muthunagai Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India

M. Nafees Muneera Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India

Athishwaran Naganathan Kumaraguru College of Technology, Coimbatore, India

Shaik Nazeer VMware Pvt. Ltd., Bengaluru, India

D. Nedumaran Central Instrumentation and Service Laboratory, University of Madras, Chennai, Tamil Nadu, India

M. Nivedita Vellore Institute of Technology, Chennai, India

Swetha Parthiban Kumaraguru College of Technology, Coimbatore, India

Sai Kiran Pasupuleti Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India

S. Poorani Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India

S. Prabhakaran SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

J. Preetharoselyn SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India

M. Premalatha Vellore Institute of Technology, Chennai, India

- P. K. Prithvi** Sri Sairam Engineering College, Chennai, India
- P. Prittopaul** Department of CSE, Velammal Engineering College, Chennai, India
- R. Priyatharshini** Easwari Engineering College, Chennai, India
- J. S. Thanga Purni** School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
- M. Radhika** Department of Information and Communication Engineering, Anna University, Chennai, Tamilnadu, India
- P. Rahul Ganesh** Easwari Engineering College, Chennai, India
- A. Raj Kumar** Easwari Engineering College, Chennai, India
- R. Ranjana** Sri Sairam Engineering College, Chennai, India
- N. Ravinder** Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India
- S. Roobini** SNS College of Technology, coimbatore, India
- R. S. Roopesh** Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, Tamil Nadu, India
- Shivanand S. Rumma** Department of Computer Science, Gulbarga University, Kalaburagi, Karnataka, India
- Sribalaji Sabanayagam** Kumaraguru College of Technology, Coimbatore, India
- R. Sandha** Thiagarajar College, Madurai, Tamil Nadu, India
- S. K. B. Sangeetha** Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India
- D. Saraswathi** Department of Computer Science, PSG College of Arts & Science, Coimbatore, India
- M. Sarath Kumar** Easwari Engineering College, Chennai, India
- K. Sarojini** Department of Computer Science, LRG Govt Arts College For Women, Tirupur, India
- K. Sasi Kala Rani** Sri Krishna College of Engineering and Technology, Coimbatore, India
- C. N. Savithri** Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, Tamil Nadu, India
- Fathima M. Shabika** Sri Sairam Engineering College, Chennai, India
- Shobana Shanmugasundaram** Kumaraguru College of Technology, Coimbatore, India

P. Shanthakumar Department of Information Technology, Kings Engineering College, Sriperumpudur, Chennai, Tamil Nadu, India

S. P. Shanthinii Department of CSE, Velammal Engineering College, Chennai, India

B. Sharmila Central Instrumentation and Service Laboratory, University of Madras, Chennai, Tamil Nadu, India

P. Sivakumar Department of Electronics and Communication Engineering, Dr. N.G.P. Institute of Technology, Coimbatore, Tamilnadu, India

D. Sowmya Department of Computer Science, PSG College of Arts & Science, Coimbatore, India

B. Sreedhar Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India

P. Sriramya Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India

T. Subha Sri Sairam Engineering College, Chennai, India

N. Suganthi Department of Computer Science, LRG Govt Arts College For Women, Tirupur, India

G. Suganya Vellore Institute of Technology, Chennai, India

M. Sujaritha Sri Krishna College of Engineering and Technology, Coimbatore, India

P. Thirumurugan Department of Electronics and Communication Engineering, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India

A. Umamageswari Department of CSE, SRM Institute of Science and Technology, Chennai, India

M. Usha Department of CSE, Velammal Engineering College, Chennai, India

M. Vasumathy The American College, Madurai, Tamil Nadu, India

R. Vedhapriyavadhana School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

L. Vijayaraja Sri Sairam Institute of Technology, Chennai, India

P. Visu Velammal Engineering College, Chennai, India

V. Viswanathan Vellore Institute of Technology, Chennai, India

Krishnan B. Yashwanth Sri Sairam Engineering College, Chennai, India

Prathyusha Yayavaram Myntra Pvt. Ltd., Bengaluru, India

B. Yogameena Department of Electronics and Communication Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India

PTZ-Camera-Based Facial Expression Analysis using Faster R-CNN for Student Engagement Recognition



E. Komagal and B. Yogameena

Abstract During the pandemic, online classes are predominated. However, the new normal needs effective analysis of students' classroom engagement. Offline classes also have a potential threat to students' engagement before and especially after the post-covid. Facial Expressions Analysis has become essential in the learning environment, whether it is online or offline. The offline classroom environment is considered a problem environment. Since, it can be easily adapted to the online environment. Notably, in the PTZ camera environment, the recognition becomes more challenging due to varying face poses, limited Field-of-View (FOV), illumination conditions, effects of the continuous pan, zoom-in, and zoom-out. In this paper, facial expression-based student engagement analysis in a classroom environment is proposed. Face detection has been achieved by YOLO (You only look once) detector to find multiple faces in the classroom with maximum speed and accuracy. Consequently, by adopting the Ensemble of Robust Constrained Local Models (ERCLM) method, landmark points are localized in detected faces even in occlusion, and therefore, feature matching is performed. Besides, the matched landmark points are aligned by an affine transformation. Finally, having different expressions, the aligned faces are fed as input to Faster R-CNN (Faster Regions with Convolutional Neural Network). It recognizes behavioral activities such as Attentiveness (Zero-In (ZI)), Non-Attentiveness (NA), Day Dreaming (DD), Napping (N), Playing with Personal Stuff in Private (PPSP), and Talking to the Students' Behind (TSB). The proposed approach is demonstrated using the TCE classroom datasets and Online datasets. The proposed framework outperforms the state-of-the-art algorithms.

E. Komagal (✉)

Department of Electronics and Communication Engineering, Latha Mathavan Engineering College, Madurai, Tamil Nadu, India

e-mail: komagale@gmail.com

B. Yogameena

Department of Electronics and Communication Engineering, Thiagarajar College of Engineering, Madurai, Tamil Nadu, India

e-mail: ymece@tce.edu

Keywords Student engagement analysis · Face expression · Pose variations · Faster R-CNN · TCE classroom dataset

1 Introduction

For more than two decades, the education system has been continuously evolving from traditional to online-based teaching and learning methods. However, in both environments, engaging students' in learning activities is challenging and tedious [1]. The research findings have revealed that the students' often misbehave in the classroom environment, disrupting the other students' learning process and the teachers' delivery. The student's behavior must be monitored and evaluated thoroughly to improve the teaching and learning environment. Therefore, behavioral studies in classroom monitoring have become the essential tool for student engagement [2, 3]. Static cameras are generally used for surveillance, but they would be too expensive (Installations to cover a larger area) or Impossible to cover a large area and zoom-in to capture facial expressions. Also, the static cameras could not detect the faces under occlusions in profile and frontal faces. Therefore, the PTZ camera is a possible solution for fulfilling these comprehensive and in-close-detail surveillance [4] needs by many security managing systems.

2 Related Work

A traditional method for Face Expression Recognition consists of four stages: Face detection, Feature extraction, Alignment, and Facial expression recognition belongs to a variety of emotions. Monitoring the students in the college environment has more facial expressions during their learning in the classroom. In the existing literature, facial expression analysis deals with emotions such as happy, excited, pleased, glad, at-ease, relaxed, calm, sleepy, miserable, sad, bored, tired, tense, afraid, and angry for various applications. The applications include face expression for action recognition, biometric, health care, surveillance, and drowsiness detection of drivers [5–9]. This paper mainly focuses on classroom environments for the students' database, which is very useful for offline as well as an online class environments.

During the past decades, mostly face detection has been widely used for recognition than facial expression, using a PTZ camera. The pioneering work of Viola and Jones [10] has initiated a face detector by cascaded Ada-Boost with Haar-like features. After that, numerous mechanism for face detection used by Wang et al. [5] and Boddeti et al. [11] encompasses on developing more advanced features and furthermore powerful classifiers. In latest years, advantages of deep learning have been more popular for face detection. So this has motivated the authors to apply deep understanding for facial expression recognition, which very few researchers have attempted. CNN-based face detectors have accomplished the highest performance.

Among the recent novel algorithms, Suryavanshi et al. and Hasani et al. [12, 13] have proposed the Fast/Faster R-CNN that executed using region-wise detections on the Regions Of Interest (ROIs). The processes investigate the contextual information for face detection and offer a structure for achieving high performance, particularly for improving small faces' correctness. Boddeti et al. [11] and Bettadapura et al. [11, 14] have proposed a single-stage summary work for face detection, through the designed approach, can realize the state-of-the-art algorithms. Among these studies, unlike other techniques, YOLO face detector utilizes the whole frame through training and testing moments and calculates the background information sequences regarding the classes and students' appearances. However, it fails to provide background patches of a frame for the region of interest objects because it cannot capture the larger environment. Redmon et al. [15] have presented a YOLO detector that performs well in detecting faces creating a few errors in the background, compared to the face detector Faster R-CNN.

Face feature extraction and alignment has constrained settings such as the absence of significant occlusions, near frontal faces, or known facial poses. The approaches proposed by Sangineto et al. [16] and Li et al. [17] attempt to find out the optimal fit of a regularized face shape model by iteratively maximizing the shape and appearance responses. However, this method has gross errors (Occlusions and Background clutter), called Outliers. Bettadapura et al. [14] and Cootes et al. [18] have addressed the shape models such as Active Shape Models (ASM) and Active Appearance Models (AAM), which are the earliest and most widely used approaches for shape fitting. Typical CLM-based methods assume that all the landmarks are visible. Ghiasi and Fowlkes [19] have proposed a CLM-based approach to account for occlusions at the learning stage by simulating facial occlusions. Unlike most previous face alignment approaches, ERCLM explicitly deals with dense occlusion and also, and this estimate also provides binary occlusion labels for each landmark and its locations. This can serve as crucial auxiliary information and, especially for this facial expression, recognition-based students' engagement recognition application.

A facial Expression Recognition (FER) system has divided interest in two major categories along with feature representations: Static frame FER and Dynamic frame FER [20, 21]. Static-based methods, Liu et al. [22] and Mollahosseini et al. [23] contain the feature representation, which predicted by means of only spatial information from the single frame. Dynamics-based methods [24, 25] used by Jung et al. [24] and Zhao et al. [25], the chronological relatives among adjacent frames facial expression order are considered. Recently, Szegedy et al. [26] has portrayed the broadly increased chip processing ability (e.g., GPU units) and well-designed network architecture, a study in a variety of fields have transferred to deep learning methods, which have accomplished the state-of-the-art identification accuracy and attained the comparative results by a more important margin. An end-to-end Identification/Recognition method using the Faster R-CNN is addressed to address the above issues. Ren et al. [27] have used Region Proposal Networks (RPNs) [27] to compute well-organized and precise regions. The pipeline uses one Convolutional Neural Network (CNN) intended for all the purposes. Therefore, the region proposal

be almost cost-free by distributing the down-flow recognition network's convolutional features [28–30]. The RPN as well improves the regional proposal importance with the accuracy of the whole target detection.

2.1 Contribution and Objective

The expression recognition is also affected by illumination, masking, aging, occlusion, and background. These issues are handled by the proposed method even in continuous pan, zoom-in, and zoom-out. The significant contribution of this paper is to use PTZ camera-based framework to recognize the students' Engagement analysis such as Attentiveness (Zero-In (ZI)), Non-Attentiveness (NA), Day Dreaming (DD), Napping (N), Playing with Personal Stuff in Private (PPSP), and Talking to the Students' Behind (TSB) using Faster R-CNN-based face expression recognition.

The remaining paper has been ordered as follows. Section 2 discusses the related works along with problem formulation, contributions, and objectives. Section 3 describes the proposed work methodology. Sections 3.1 and 3.2 depicts the YOLO Detector and ERCLM face landmark points, and then Sect. 3.3 focuses on the alignment and transformation. Section 3.4 defines face recognition by using Faster R-CNN. The results and discussion are discussed in Sect. 4, and then Sect. 4.1 describes performance measures. Finally, a conclusion is given in Sect. 5.

3 Methodology

In this paper, a framework and stages are shown in Fig. 1. To develop and to determine a facial expression for student engagement in the active learning environment. Here, face detection has been achieved by the YOLO detector to find multiple faces in the classroom with maximum speed and accuracy. Subsequently, implemented by the Ensemble of Robust Constrained Local Models (ERCLM) method, the landmark points are received and localized in detected faces even in occlusion feature matching, performed on the detected features. An affine transformation aligns them for fitting the face. Finally, the aligned faces have different expressions fed as input into Faster R-CNN. It recognizes behavioral activities such as Attentiveness (Zero-In (ZI)), Non-Attentiveness (NA), Day Dreaming (DD), Napping (N), Playing with Personal Stuff in Private (PPSP), and Talking to the Students' Behind (TSB). An evaluation of subject-independent analysis is experimented with using the Online datasets and TCE datasets created on our college campus. The proposed methodology has demonstrated the best output score performance in recognizing facial expressions for behavioral studies. It has also been tested in a continuous pan, zoom-in, and zoom-out scenario to validate even in pose variations.

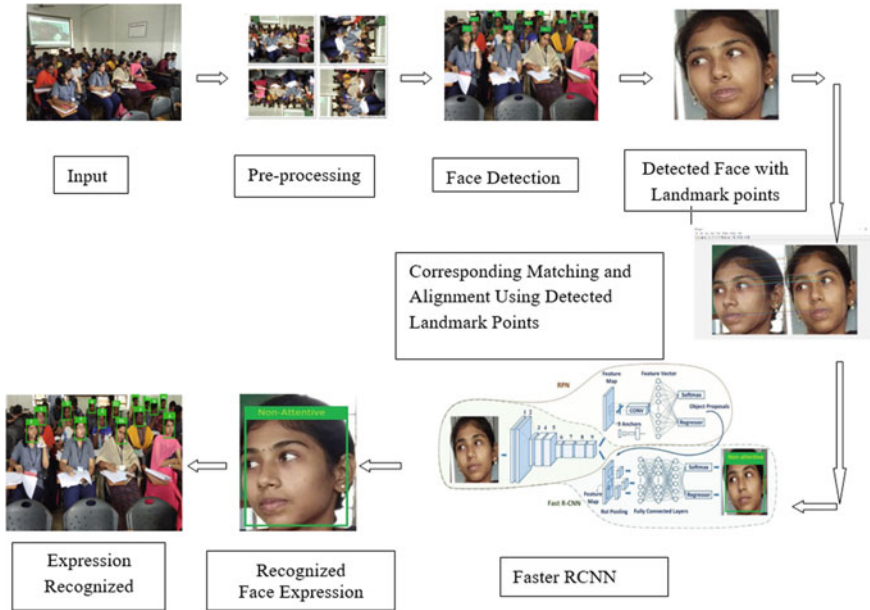


Fig. 1 Proposed framework on facial expression recognition-based student engagement analysis

3.1 Face Detection using YOLO Detector

YOLO detector makes a fewer number of background errors because of its faster prediction with valid inputs when compared to Fast R-CNN. Thus this method is used for face detection. Input Videos are given for detecting faces in the classroom scenario, and the faces are seen in real time with much precision. The YOLO face detection classification re-size the face expression frame to 448×448 , executed a single convolutional network, and then thresholds the face expression detection through the model's confidence. YOLO unifies face detection into a single neural network. This network utilizes features of the whole frame to calculate every bounding box. The network calculates and detects the faces globally. The YOLO design facilitates end-to-end validation in real-time and maintains the speeds by high standard precision. YOLO splits the input face expression frame into $S \times S$ grid. If the middle point of a face expression falls into a grid cell, that grid cell is designed for identifying that particular face. All grid cell predicts B bounding boxes through confidence scores. These confidence scores point to the frame and reveal that the box holds a natural look of that face. It defines confidence as Eq. (1).

$$P_r(\text{Class}_i | \text{Object}) * P_r(\text{Object}) * IOU_{Pred}^{Truth} = P_r(\text{Class}_i) * IOU_{Pred}^{Truth} \quad (1)$$

where Intersection Over Union (IOU) among the predicted box and the ground truth. The confidence scores have to be zero when there is no face exists. Otherwise, the

confidence score must be equivalent to the Intersection Over Union (IOU) between the predicted box and the ground truth. All bounding box frames have the five predictions: x , y , w , h , and confidence. The (x, y) coordinates symbolize the middle of the box in relation to the grid cell's bounds. The width and height are calculated as close to the complete frame. At last, the confidence predictions stand for the IOU among the predicted box and any ground truth box. YOLO detector makes fewer background errors when compared to Fast R-CNN. Input videos are given for detecting faces in a classroom scenario, and the faces are detected in real time with much precision.

3.2 *Detection of Landmark Points using Ensemble of Robust Constrained Local Models (CLM)*

The framework of ERCLM has been elaborated on the following reasons for finding the dense landmark points. Independent shape and landmark appearance models are used to obtain the discretized representation of pose, expression, and binary occlusion labels. Given the various inputs of facial poses, a region of expressions, and a shape model, the corresponding local landmark detectors across different orientations are used to acquire the exact response maps. The response maps extract landmark locations for the consequent shape model. The detected landmarks points are extracted from the input.

3.3 *Affine Transformation*

The 2D affine transformation is used for face alignment across various poses. The change is represented, as depicted in Eqs. (2) and (3):

$$x_n = S_x \cdot \text{Cos}\theta \cdot x_o - S_x \cdot \text{Sin}\theta \cdot y_o + (x_r + t_{xr} - S_x \cdot \text{Cos}\theta \cdot x_r + S_x \cdot \text{Sin}\theta \cdot y_r) \quad (2)$$

$$y_n = S_y \cdot \text{Sin}\theta \cdot x_o + S_y \cdot \text{Cos}\theta \cdot y_o + (y_r + t_{yr} - S_y \cdot \text{Sin}\theta \cdot x_r - S_y \cdot \text{Cos}\theta \cdot y_r) \quad (3)$$

Here, (x_o, y_o) , (x_n, y_n) is old. The new coordinates after the transformation are the center of rotation, θ is the rotation angle in the X-axis and Y-axis translations according to the center of the process, index S_x and S_y represent the size scales of X-axis and Y-axis, respectively. After landmark point correspondences, the two frames which have been matched are warped. Then, the warped frame fed an input for facial expression recognition.

3.4 Face Expression Recognition using Faster R-CNN (Faster Regions with Convolutional Neural Network)

Faster R-CNN (Faster Regions with Convolutional Neural Network) is implemented for facial expression recognition more efficient to avoid the difficult explicit feature extraction method and the complexity of low-level information concerned in traditional facial expression recognition. Initially, the facial expression frame be used for normalization, in addition to the inherent features be taken out through the trainable convolution kernel. Then, the maximum pooling is familiar to diminish the dimensions of the extracted implicit features. After that, RPNs (Region Proposal Networks) produce the high-quality region proposals employed through Faster R-CNN for detection. It includes Layers (C1, C3), max-pooling layer (M2), top two layers (F7 and F8), which be subsequent layers (L4, L5, and L6), and fully connected. The completely connected layer is designed for K-way softmax, which splits the various class labels (where K is the number of classes). Finally, the softmax classifier and regression layer categorize facial expressions and calculate the test sample's boundary box, correspondingly. These labels are used to recognize various facial expressions. This Faster R-CNN outperforms effectively than CNN and Fast CNN. Since Faster R-CNN has been used, each frame's Region Of Interest (ROI) has to be distinct first, accomplished by the coordinates of ROI. At that moment, the coordinates are transferred to XML format.

4 Results and Discussion

The proposed work is performed using MATLAB 2018b and experimentation done by on various classroom datasets. The specifications include classroom monitoring datasets, as given in Tables 1 and 2. Online datasets (2018) and TCE datasets (2019) with multiple expressions in the classroom are taken for experimentation, as shown in Table 3. These datasets contain faces in a controlled environment condition with varying viewpoints, occlusion, and illumination. The benchmark datasets taken from the (TCE) college dataset using the PTZ camera are furthermore used in this experimentation. Training frames of students for facial expression taken from Online and TCE college datasets. These two samples (Online and TCE dataset) have been used for training and testing. Examples of phrases under challenging conditions are Attentiveness (Zero-In (ZI)), Non-Attentiveness (NA), Day Dreaming (DD), Napping (N), Playing with Personal Stuff in Private (PPSP), and Talking to the Students' Behind (TSB) for Various Expressions in Continuous pan, Excess zoom-in and zoom-out conditions with different poses from TCE Datasets are shown in Table 1.

At first, the face detection is through the YOLO detector, which detects all faces in the classroom and puts bounding boxes of every face. It is illustrated on TCE and Online datasets, as shown in Table 3, work well for complete profiles and fully occluded faces with the advantage of its speediness. Subsequently, the detected faces

Table 1 Training images of TCE students' for Facial Expression Recognition




Various Expressions\View	Training images for seven different Facial Expression Recognition
Attentiveness (Zero-In (ZI)) (Frontal)	
Attentiveness (Zero-In (ZI)) (Profile)	
Non-Attentiveness (NA) (Profile)	
Napping (N) (Profile)	
Talking to the Students' Behind (TSB) (Profile)	
Day Dreaming (DD) (Frontal)	
Playing with Personal Stuff in Private (PPSP) (Profile)	

Table 2 Classroom monitoring datasets and their specification

Dataset description	Online datasets (2018)	TCE datasets (2019)
Frames		
Total Frames	1309	20,641
Resolution	640 × 360	704 × 288
Challenging Conditions	Target size, occlusion of the face with objects like pen and hand	Shadow, Illumination Wearing goggles
View angle	Profile	Frontal
Frame format	.jpg	.jpg
Frames per second	25fps	24fps

are used to obtain the landmark points using ERCLM. Localized landmark points (fiducial points) of ERCLM extract dense landmark points specially deal with various expressions as illustrated in Fig. 2. Consecutive frames of landmark points (fiducial points) are used to match their corresponding landmark points, as shown in Fig. 3. And then, the matched landmark points are aligned by an affine transformation, as

Table 3 Students' Face Expressions in classroom

Continuous pan, Excess zoom-in and Excess zoom-out					
Online Datasets (2018)			TCE Classroom Datasets(2019)		
Original Frame 20	Original Frame 35	Original Frame 45	Original Frame 15	Original Frame 5	Original Frame 12
Continuous Pan	Excess Zoom-In	Excess Zoom-Out	Continuous Pan	Excess Zoom-In	Excess Zoom-Out
YOLO Detection of Faces in Classroom					
Faster R-CNN for Faces in Classroom					

shown in Fig. 4. Finally, the aligned pose faces are given for face expression recognition by Faster R-CNN. It classifies the expression and their validated results are mentioned in Tables 4 and 5 for the Online datasets and TCE datasets.



Fig. 2 Various Facial Expressions and their ERCLM landmark points



Fig. 3 Various Facial Expressions and the correspondence of matching points

Fig. 4 Face alignment across various poses



4.1 Performance Analysis

The following performance metrics are used for validation of the experimental results obtained by the proposed framework as shown in Tables 4 and 5. They are given in the following Eqs. (4–7). Accuracy of recognised faces is defined in Eq. (4),

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

where True positive (TP) = Number of face expressions identified correctly, True negative (TN) = Number of face expressions not able to be identified when those expressions are not present, False positive (FP) = Number of falsely identified face expressions, False negative (FN) = Number of face expressions identified when those expressions are not present. Precision, otherwise positive prediction range is distinct as the amount of the true positives against all positive results as shown in Eq. (5).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

Detection Rate provides a % of face expressions correctly identified is given in Eq. (6).

$$\text{Detection Rate} = \frac{TP}{\text{Total Number of face expressions}} \quad (6)$$

False Detection Rate demonstrates the % of face expressions falsely identified is given in Eq. (7).

$$\text{False Detection Rate} = \frac{FP}{\text{Total Number of face expressions}} \quad (7)$$

Table 4 Performance measures of Online Datasets for Students' Facial Expression Recognition













Databases	Faces	TP	FP	TN	FN	Accuracy	Detection Rate (%)	Precision (%)	False Detection Rate (%)
Attentiveness (Zero-In (ZI))		90	20	80	20	80.952	90	81.8182	20
Non-Attentiveness (NA)		95	32	86	40	71.542	95	74.8031	32
Day Dreaming (DD)		90	26	90	32	75.630	90	77.5862	26
Napping (N)		90	28	92	35	74.286	90	76.2712	28
Playing with Personal Stuff in Private (PPSP)		92	30	86	56	67.424	92	75.4098	30
Talking to the Students' Behind (TSB)		93	26	90	56	69.057	93	78.1513	26

Table 5 Performance measures of TCE Datasets for Students' Facial Expression Recognition

Databases	Faces	TP	FP	TN	FN	Accuracy	Detection Rate (%)	Precision (%)	False Detection Rate(%)
Attentiveness (Zero-In (ZI))		80	20	80	20	80	13.33	80	3.33
Non-Attentiveness (NA)		85	32	86	40	70.37	14.16	72.64	5.33
Day Dreaming (DD)		80	26	90	32	74.56	13.33	75.47	4.33
Napping (N)		80	28	92	35	73.19	13.33	74.07	4.66
Playing with Personal Stuff in Private(PFSP)		82	30	86	56	66.14	13.66	73.21	5
Talking to the Students' Behind (TSB)		83	26	90	56	67.84	13.83	76.14	4.33

5 Conclusion

An application framework, such as face expression-based students' engagement analysis for the offline and online class environment, has been proposed. Face Expression Recognition is very challenging due to illumination conditions, degree of rotation of the face. Students various expressions in the classroom are monitored with the PTZ camera. The environment has been tested under the continuous pan, zoom-in, and zoom-out. ERCLM, which provides dense land marking schemes, predominantly deals with occlusion. Feature corresponding landmark points are matched and aligned by an affine transformation, and these results are used for recognition by Faster R-CNN. Thus, Faster R-CNN recognizes various students' expressions such as Attentiveness (Zero-In (ZI)), Non-Attentiveness (NA), Day Dreaming (DD), Napping (N), Playing with Personal Stuff in Private (PPSP), and Talking to the Students' Behind (TSB). Dataset collection for this application is one of the contributions since very few datasets are only available publicly. Various datasets of students' expressions taken from TCE college datasets (2019) and Online datasets (2018) are tested, and the proposed framework recognizes face expressions for various pose variations. The performance analysis of facial expression has higher accuracy for frontal faces, low detection time, high recognition rate; using the proposed techniques, it is simple and faster. Hence, it can be easily adapted for the online learning environment. In the future, more behavior in expressions has been done.

References

1. Sun RCF, Daniel TL, Shek (2012) Student classroom misbehavior: an exploratory study based on teachers' perceptions. *Sci World J* <https://doi.org/10.1100/2012/208907>
2. Khalfallah J, Slama JBH (2015) Facial expression recognition for intelligent tutoring systems in remote laboratories platform. *Procedia Computer Sci* 73: 274–281
3. Michael Revina I, Sam Emmanuel WR (2021) A survey on human face expression recognition techniques. *J King Saud Univ Comput Inf Sci* 33(6):619–628
4. Zhang J, Wang Y, Chen J, Xue K (2010) A framework of surveillance system using a PTZ camera. In: 3rd international conference on computer science and information technology, IEEE
5. Wang N, Gao X, Tao D, Li X (2014) Facial feature point detection: A comprehensive survey. In: *Computer Vision and Pattern Recognition*
6. Ranjan R, Sankaranarayanan S, Castillo CD, Chellappa R (2017) An All-In-One convolutional neural network for face analysis. In: 12th IEEE international conference on automatic face & gesture recognition
7. Huang Y, Chen F, Lv S, Wang X (2019) Facial expression recognition: a survey. *Symmetry* 11:1189. <https://doi.org/10.3390/sym11101189>
8. Samadiani N, Huang G, Cai B, Luo W, Chi C-H, Xiang Y, He J (2019) A review on automatic facial expression recognition systems assisted by multimodal sensor data. *Sensors* 19(8):1863. <https://doi.org/10.3390/s19081863>
9. Ashwin TS, Guddeti RMR (2020) Affective database for e-learning and classroom environments using Indian students' faces, hand gestures and body postures. *Future Gener Comput Syst* 108:334–348

10. Viola P, Jones M (2004) Robust real-time face detection. *Int J Comput Vis* 57(2):137–154
11. Boddeti VN, Roh MC, Shin J, Oguri T, Kanade T (2017) Face alignment robust to pose, expressions and occlusions. In: *Computer Vision and Pattern Recognition (CVPR)*
12. Suryavanshi SR, Sankpal LJ (2016) Monitor student's presence in classroom. *J Inf Technol Softw Eng* 6. <https://doi.org/10.4172/2165-7866.1000185>
13. Hasani B, Mohammad H, Mahoor (2017) Facial expression recognition using enhanced deep 3D convolutional neural networks. In: *Computer Vision and Pattern Recognition*. [arXiv:1705.07871v1](https://arxiv.org/abs/1705.07871v1) [cs.CV]
14. Bettadapura V (2012) Face expression recognition and analysis: the state of the art, *Computer Vision and Pattern Recognition (CVPR)*
15. Redmon J, Farhadi A (2016) YOLO9000: better, faster, stronger. In: *Computer Vision and Pattern Recognition (CVPR)*
16. Sangineto E (2013) Pose and expression independent facial landmark localization using Dense-SURF and the Hausdorff distance. *IEEE Trans Pattern Anal Mach Intell* 35(3)
17. Li X, Xu Y, Lv Qi, Dou Y (2016) Affine-transformation parameters regression for face alignment. *IEEE Signal Process Lett* 23(1)
18. Cootes TF, Edwards GJ, Taylor CJ (2001) Active appearance models. *IEEE Trans Pattern Anal Mach Intell (PAMI)* 23(6):681–685
19. Ghiasi G, Fowlkes CC (2014) Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp 1899–1906
20. Yenugaladoddi Jayasimha R, Reddy VS (2021) A facial expression recognition model using hybrid feature selection and support vector machines. *Int J Inf Comput Secur* 14(1):79–97
21. Canedo D, António JR, Neves (2019) Facial expression recognition using computer vision: a systematic review. *Appl Sci: Object Detection using Deep Learning for Auton Intell Robots* 9(21):4678. <https://doi.org/10.3390/app9214678>
22. Liu P, Han S, Meng Z, Tong Y (2014) Facial expression recognition via a Boosted Deep Belief Network. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 1805–1812
23. Mollahosseini A, Chan D, Mahoor MH (2016) Going deeper in facial expression recognition using deep neural networks. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp 1–10
24. Jung H, Lee S, Yim J, Park S, Kim J (2015) Joint fine-tuning in deep neural networks for facial expression recognition. In: *IEEE International Conference on Computer Vision (ICCV)*, pp 2983–2991
25. Zhao X, Liang X, Liu L, Li T, Han Y, Vasconcelos N, Yan S (2016) Peak-Piloted deep network for facial expression recognition. In: *European Conference on Computer Vision (ECCV)*, *Lecture Notes in Computer Science*, vol 9906. Springer, pp 425–442. https://doi.org/10.1007/978-3-319-46475-6_27
26. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 1–9
27. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: *NIPS'15: Proceedings of the 28th international conference on neural information processing system*, vol 1, pp 91–99
28. Wang Y, Ji X, Zhou Z, Wang H, Li Z (2017) Detecting faces using region-based fully convolutional networks. In: *Computer Vision and Pattern Recognition (CVPR)*
29. Li J, Zhang D, Zhang Ji, Zhang Ju, Li T, Xia Yi, Yan Q, Xun L (2017) Facial expression recognition with Faster R-CNN. *Procedia Comput Sci* 107:135–140
30. Xu L, Fei M, Zhou W, Yang A (2018) Face expression recognition based on Convolutional Neural Network. In: *Australian & New Zealand Control Conference (ANZCC)*, IEEE

Convergence Perceptual Model for Computing Time Series Data on Fog Environment



Rupa Kesavan , S. Poorani, R. Iyswarya, S. U. Muthunagai, R. Anitha,
and L. Vijayaraja 

Abstract The evolution of fog/edge paradigm is at the rising edge that complements data analysis and data computing algorithms in a big data platform, The role of fog nodes in a fog computing set up shall best suit time sensitive applications especially on virtual clusters supporting edge devices in sensing, processing, controlling and action planning, there by replacing the non-virtualized complicated computing mechanisms resolving the existing complexity in cloud computing storage and retrieval algorithms. There is an emerging need for addressing collaborative processes in various sectors not limiting to computing digital data on big data platform but also the futuristic industrial revolution to replace the existing techniques and technologies in industrial automation. The proposed work reveals a novel framework that supports the cutting edge of fog node fitting as an intermediate layer between the edge devices and cloud storage. Computational by means of fog-node attracts the need by replacing existing cloudlets time complexity in resource management. Hence fog as things operates as a controller in cyber physical systems like controlling transmission lines of high volt system connections, monitoring smart metering that has a wide range of applications through the integration of the industrial automation process.

Keywords Time series applications · Fog computing · Horizontal architecture · Fog as things · Perceptual model · Fog architecture · Fog for industry 4.0

R. Kesavan (✉) · S. Poorani · R. Iyswarya · S. U. Muthunagai · R. Anitha
Sri Venkateswara College of Engineering, Pennalur, Sriperumbudur, India
e-mail: rupakesavan@svce.ac.in

L. Vijayaraja
Sri Sairam Institute of Technology, West Tambaram, Chennai, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
R. J. Kannan et al. (eds.), *Computer Vision and Machine Intelligence Paradigms
for SDGs*, Lecture Notes in Electrical Engineering 967,
https://doi.org/10.1007/978-981-19-7169-3_2

1 Introduction

In this proposed model a convergent architecture uses fog nodes to improve the bandwidth and reduce latency for content delivery on the cloud environment targeting the edge devices in emerging areas not limiting to vehicular ad hoc network for computing time-series open data sets [1]. In [2] the computing environment manages the resources on the open source which increases the latency time in data management whereas the utilization of fog nodes is recommended to manage the IoT resources and allow cloud platforms to act according to the current situations of end-users connected to edges.

The process of data management includes scheduling competently, stimulating QoS, allocation, and managing of cloud resource, valuing, and security. Reference [2] evaluates the cloud resources necessary for processing requests according to the behavior and historical record of users [3], also controlling autonomous vehicle system, resource management in electric vehicle, power drives, real-time variable motor control, and smart grid integration systems (this leads to minimized cloud resources under utilization [4]). A systematic mechanism for pre-processing the data before sending it to the cloud by storing it in the fog nodes, leading to mobility and invulnerability is provided here. In the case of airborne surveillance system, authorizing the services through high-performance global navigation sensors [GNSS] equipped with improved performance in object detection and tracking the services thus ensuring the docking services relying on the end devices interfaced to fog nodes.

1.1 Goals Aimed at the Convergence Fog Model

To provide solutions for the challenges in VANETs [4], by providing better connectivity, adjustability, pliability, and intelligence decision computation through inter-fog node transmission for secured communication [5] is proposed.

A computing paradigm focuses on the industrial devices connecting smart cities that limits the object tracking using static sensors and deployed on a private cloud that has limited bandwidth and moderate latency [5]. Fog to cloud interaction also slows to communicate with each other, and it reduces the overall end-to-end delay.

1.2 Fog for Time-Series Computing

The proposed model provides modern user interface enriching node services by bringing a powerful changes in industrial and vehicular convergence models, the existing vertical architecture has top to bottom approach in networking models including a cloud platform that has complexity with data access with limited

bandwidth for the heterogeneous edges. Thus, the improved model emerges with horizontal architecture connecting homogeneous level devices interfaced to the heterogeneous computing platform by locally manipulating the data.

Fog as an Evolving Technology. The value of fog emerges enriches analytics capability replacing manual operations, static functionalities in computing algorithms to take quick decisions, in preventing quality degradation, and performance drops, in huge implications, thus analytics of fog play a vital role in predictive implications, predictive maintenance in a layered approach.

Proposed Layered Approach-Convergence Perceptual Model [CPM]

Fog aims at providing optimal latency with improved bandwidth for automated systems in object detection and tracking by perceptual manifestation of data, this can be accomplished by locally managing data control for high end Airborne surveillance systems, submarines, navigation controls in sailings, etc., Thus improved fog computing mechanisms facilitates expanded fragmentation controls on various hardware's interfaced with the edge devices but necessarily to locate the data on the primary cloudlet.

Figure 1 includes a horizontal view of the proposed architecture, a complete model works on the virtualization technique hosted on cloud platform, the physical layer is the one where the edge devices are hosted with the improved location-based sensors [6], global navigation sensor system [GNSS], supporting Inertia measuring unit [IMU], Radar sensor, etc., the edge device in the proposed system facilitates any device that generates time-series data especially the device oriented with vehicular Adhoc networking, the system targets data management features perceiving raw data from any of the equipped high processing sensors.

The storage layer and pre-processing layer involve data integration with fog nodes, data supervisory control from the cloudlets is interfaced through a horizontal medium where virtual cluster nodes are inter-connected with at least one fog node [7], Fog nodes allow local data manipulation.

The learning model can be any of the machine learning approach to train the data from the edge devices, data pre-processing includes perceiving data from the edge device and localizing the edges on the virtual clusters, the efficacy of processing node ensures reporting data from the cluster node to the fog node without report failure [8]. The existing computing platform involves complexity in processing cluster nodes with a single cloudlet, since the unique cloud server hosts the entire computation, the bandwidth and latency period of the data computation is relatively high which degrades the performance of cluster nodes. Hence a heterogeneous and convergence model demands localization achieving high data reporting remarkably docking services with reduced delay in response.

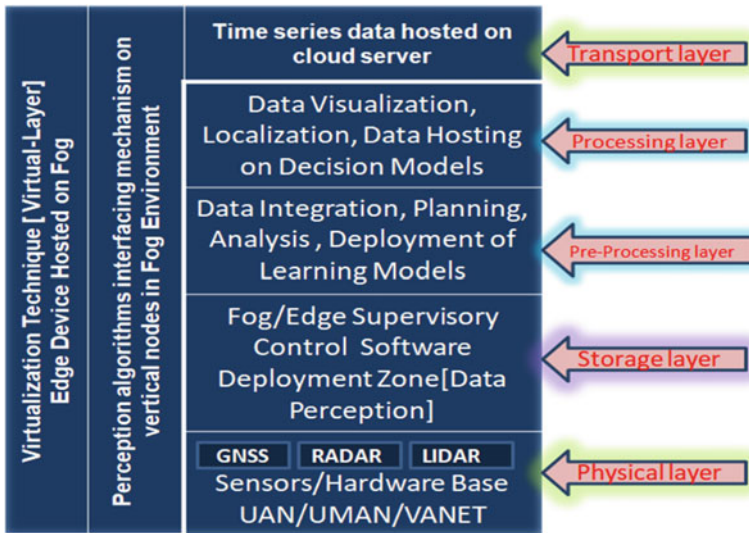


Fig. 1 Layered architecture of convergence perceptual model

1.3 Proposed Convergence Perceptual Architecture

Figure 2 is a tri-layered architecture model that complements the existing distributed hierarchical approach, here the computing paradigm focus on the industrial devices connecting smart cities [9] that limit the object tracking using the static sensors, and deployed on a private cloud that has limited bandwidth and moderate latency. Whereas the proposed model shall be implemented on the heterogeneous platform with edge devices equipped with multiple improved sensors ensuring nil failure in reporting data at an optimal response time when compared with other services on a computing platform.

1.4 Perceptual Layer on CPM

The perceptual layer in Fig. 2 ensures localization achieved with a decomposable data analytics operating technology using a learning model [10] in fog computing to reduce the latency time, the proposed model assures any machine learning model synchronized with data perception for local computing mechanism which supports fragmentation at fog nodes through decomposition technique. Hence Trade-offs in cost, latency, and quality of results are relative to higher demand for the horizontal architecture integrated with the heterogeneous networking framework on the convergence model.

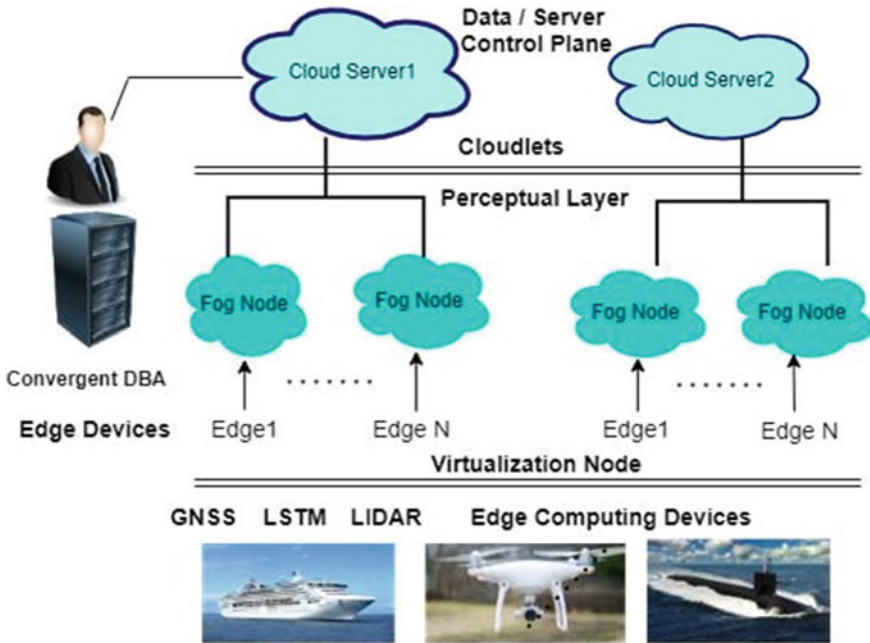


Fig. 2 Architecture of convergence perceptual model illustration of fog nodes for cyber physical systems interfaced edges on a cloud-based environment

Data Integration at the pre-processing layer on the computing networking platform is possible by means of fragmenting the data analysis through time-series data perceived at the fog nodes, since local manifestation of data is operated at customized sensor modes, hence the predictable data rate is attained at the cloudlet server.

In [11], the author proposed a horizontal tri-layered architecture to transform the computing mechanism realized at the enterprise level is remarkable through the perceptual mechanism for the various fog nodes interfaced with the edge devices and the multiple cloudlet servers. This strategy facilitates edge devices connected to a distinguished platform that can be interfaced on the virtual zone during local computation means that attracts the twining of cyber physical objects to locally manipulate the industrial operations of monitoring, adjusting, and managing voltage level inputs at variable speed motor drives in the industrial operations, similarly manifesting smart controlling operations at high voltage motor end to avoid voltage drops [12].

Simulation Platforms and Evaluation of Performance

Evaluation of variable inputs can be realized through any Open CV, dialogflow, and graph databases to manipulate time-series data on the cloud/fog environment. Simulation of cluster nodes shall be made using MATLAB [10], Game Engine Tools, Blender software-defined models to evaluate the interaction of cyber physical systems to collaborate with the active nodes in the fog environment.

A comparative study is made on the performance analysis for varying network bandwidth and latency on multiple cloud platforms for time-series devices. In [13] fog computing Era demands addressing the horizontal approach for managing resources in a distributed computing environment, hence a reliable service model that overcomes the time complexity in a cloud environment is required.

Data management on a horizontal approach that processes raw data from the superimposed sensors operating at a high-frequency bandwidth with an optimal speed retrieving the necessary data rather than computing the historical data, thus the data management for a heterogeneous computing platform is highly reliable.

Rather cross-fog nodes shall also be evaluated to verify the authorized nodes for the promising fog node services to ensure the optimal services of local computation of the cloudlet clusters in a big data platform.

More over the performance analysis is noteworthy for the emerging industrial automation of twining Fog-of- things with the realization of smart computing services of assembled elements in industrial 4.0/industrial 5.0 using the 5G technology for effective latency and bandwidth simulated under any software-defined simulators/emulators in futuristic applications like smart metering, smart inverter technology [14], renewable energy resources, smart grid computations, etc.

Table 1 shows a comparison of varied applications that proves the latency time deployed over the cloud platform and using the fog as a service at the varying time at varying frequencies.

Future Challenges

Open Challenges: It is observed that the provision of multiple fog nodes across the varying server interfacing necessary fog nodes only to the master node will have better efficiency on the open data set architectures that will have great performance toward the optimal latency to compute complex system at a better response time.

Digital transformation of several functional areas for industrial automation requires the computation of real-time [time series] data that involves intensive computation using the intelligent algorithms on the cloud platform, hence the integration of promising nodes through local computation shall bring major changes through non-exhaustive services in the smart factories like transmission lines, manufacturing sectors, atomic plant units, etc.

Table 1 Latency performance on cloud vs fog environment

Applications	Sensors	Operating frequency	Bandwidth	Observed latency on cloud	Predicted latency on Fog
Hazard Alert	WSN UWB Ultra sound vision-based sensing	3.1 to 6 Hz	<1 Mbps	10 ms	~7 ms
Automotive e-Call	GPS	1575.42 MHz	<1 Mbps	100 ms	~50 ms
Autonomous Driving	RADAR LIDAR	400 MHz–36 GHZ 200 THZ	10 Mbps	1 ms	<0.001 ms
Real-Time Gaming	MEMS EEG	77 GHZ–94 GHZ 13 Hz–30 Hz	10 Mbps	10 ms	<3 ms
Wireless cloud-based office	PPG	0.5–4.0 Hz	100 Mbps	1000 ms	<100 ms

Fog may serve as the primary role in resource optimization by exploring promising services for the trusted, reliable services through Node management by means of perceptual node services identifying the time-series analysis on the cloud platform.

Hence the proposed model leaves the above ideas for the future works that shall be carried out to investigate on enhanced technologies to reserve primary data set at the local nodes for processing algorithms and to have a unique framework for data set managed on the fog node shall be carried out to prove the computing efficiency.

2 Conclusion

As the Internet of things has become an integral part of technology in everything structured and unstructured data in Tons, zetta byte, rate of data growth is unpredictable in industrial growth that requires, new exponential computing mechanisms for better performance in industry 4.0 for breakthrough technologies, hence this can be well complemented by the proposed architecture layers by investigating computing means as well as storage means for better efficiency and latency shown in Table 1 and Fig. 3.

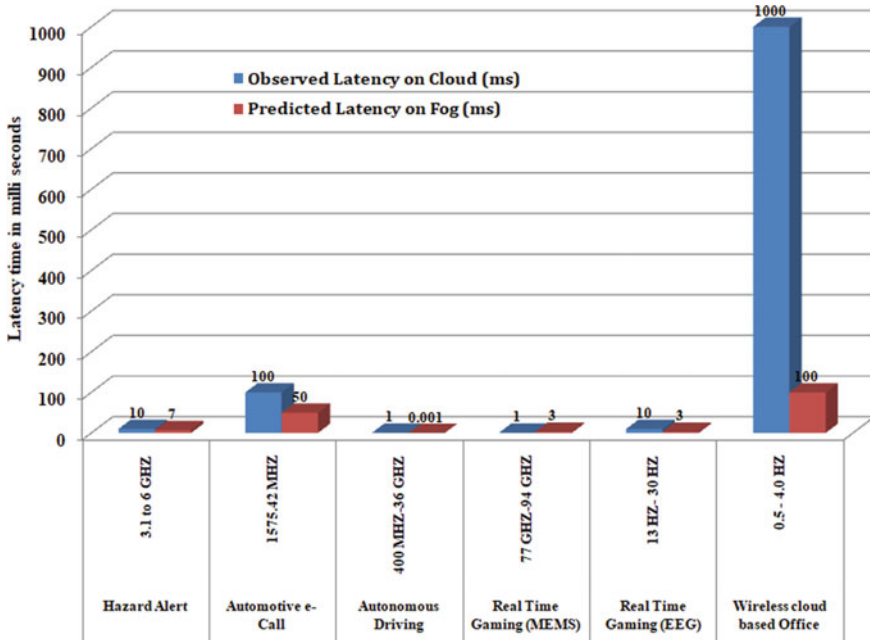


Fig. 3 Performance analysis of latency on edge devices in a cloud versus fog environment

References

1. Abdulkareem KH et al (2019) A review of fog computing and machine learning: concepts, applications, challenges, and open issues. In: IEEE Access, vol 7, pp 153123–153140
2. Mukherjee M et al (2018) Survey of fog computing: fundamental, network applications, and research challenges. IEEE Commun Surv Tutorials 20:1826–1857
3. Tang B, Chen Z, Hefferman G, Wei T, He H, Yang Q (2015) A hierarchical distributed fog computing architecture for big data analysis in smart cities. In: Proceedings of the ASE Big data & social informatics
4. Kamal MB, Javaid N, Ali Naqv SA (2019) Heuristic min-conflicts optimizing technique for load balancing on fog computing. Advances in intelligent networking and collaborative systems, vol 23
5. Kesavan R, Vijayaraja L (2017) A key management technique for cloud storage using semi trusted policy preservation. Int J Creative Res Thoughts 5:3306–3319
6. Ajay Krishnan R, Magesh kumar P, Vignesh R, Kesavan R, Veera Lakshmi P (2018) An automated design for ATC (Air Traffic Control) for airport monitoring system using TORADEX single board computer. Int J Res Appl Sci Eng Technol 6:2259–2267
7. Kesavan R et al (2021) A cloud-based user recommender and e-governing system for public welfare. IVCRAISE2020, IOP conference series: mater. science engineering, 1055 012062
8. JOR, Wanchun D, Jiang J, Li Z, Tian Y, Al-Nabhan N (2020) A review of techniques and methods for IoT applications in collaborative cloud-fog environment. Secur Commun Netw Hindawi, 1939–0114. <https://doi.org/10.1155/2020/8849181>
9. Pop P, Zarrin B, Barzegaran B, Schulte S, Punnekkat S, Ruh J, Steiner W (2021) The FORA fog computing platform for, industrial IoT. Inform Syst, vol 98. 101727,ISSN0306–4379, <https://doi.org/10.1016/j.is.2021.101727>

10. Svorobej S, Takako Endo P, Bendeache M, Filelis-Papadopoulos C, Giannoutakis KM, Gravvanis GA, Tzovaras D, Byrne J, Lynn T (2019) Simulating fog and edge computing scenarios: an overview and research challenges. *Future Int* (11). <https://doi.org/10.3390/fi11030055>
11. Kesavan R, Loganathan V, Shankar T, Periasamy JK (2022) Fog-computing: a novel approach for cloud-based devices using perceptual cloning manifestation-PerColNif taxonomy by energy optimization. In: Tiwari R, Mittal M, Goyal LM (eds) *energy conservation solutions for fog-edge computing paradigms. Lecture Notes on Data Engineering and Communications Technologies*, vol 74. Springer, Singapore. https://doi.org/10.1007/978-981-16-3448-2_6
12. Loganathan V, Srinivasan GK, Rivera M (2020) Realization of 485 level inverter using Tri-state architecture for renewable energy systems. *Energies* 13(24):6627. <https://doi.org/10.3390/en13246627>
13. Adams F (2017) OpenFog reference architecture for fog computing. <https://knect365.com/Cloud-enterprise-tech/article/0fa40de2-6596-4060-901d-8bddd167cfe/openFog-reference-architecture-for-Fog-computing>
14. Srinivasan GK, Rivera M, Loganathan V, Ravikumar D, Mohan B (2021) Trends and challenges in multi-level inverter with reduced switches *electronics* 10(4):368. <https://doi.org/10.3390/electronics10040368>

Localized Super Resolution for Foreground Images Using U-Net and MR-CNN



Umashankar Kumaravelan and M. Nivedita

Abstract Images play a vital role in understanding data through visual representation. It gives a clear representation of the object in context. But if this image is not clear it might not be of much use. Thus, the topic of image super resolution arose and many researchers have been working toward applying computer vision and deep learning techniques to increase the quality of images. One of the applications of super resolution is to increase the quality of portrait images. Portrait images are images that mainly focus on capturing the essence of the main object in the frame, where the object in context is highlighted whereas the background is occluded. When performing super resolution the model tries to increase the overall resolution of the image. But in portrait images the foreground resolution is more important than that of the background. In this paper, the performance of a Convolutional Neural Network (CNN) architecture known as U-Net for super resolution combined with Mask Region-Based CNN (MR-CNN) for foreground super resolution is analyzed. This analysis is carried out based on localized super resolution, i.e., we pass the LR Images to a pre-trained image segmentation model (MR-CNN) and perform super resolution inference on the foreground or segmented images and compute the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) metrics for comparisons.

Keywords Super resolution · Deep learning · U-Net · Mask-RCNN

U. Kumaravelan (✉)
Ydealogy Ventures, Chennai, India
e-mail: umashanks99@gmail.com

M. Nivedita
Vellore Institute of Technology, Chennai, India
e-mail: nivedita.m@vit.ac.in

1 Introduction

The challenging task of computing a high-resolution image from a single low resolution has given rise to multiple state-of-the-art architectures. Many researchers are still working towards new methods and architectures to increase the performance of previous models. Many of the papers aim at improving or modifying the model architecture or the loss and optimizer functions to increase performance. The influence of the image region also plays a vital role in super resolution since the main goal is to increase resolution such that the object or content in context is more clearly visible. Images such as portrait images and passport photos all have a common characteristic, i.e., the foreground is clear, whereas the background is blurred or occluded. For such images, quality is a crucial aspect, and it is the foreground resolution that matters. In this work, we propose a combination of image segmentation models for foreground super resolution. The proposed model is a U-Net [1] based CNN architecture with MR-CNN [2] for computing the super resolution output of the foreground or segmented image. The output of the MR-CNN model, i.e., the segmented super resolution result, is added back to the LR image. Then the evaluation metrics such as SSIM and PSNR between the Segmented Super Resolution (SR) output and normal SR output are computed to evaluate the increase in metric performance. In this paper, we shall go through some of the architectures and techniques involved in super resolution. Then we will go through our proposed architecture and its implementation details, followed by the results and observations we had come across.

2 Literature Survey

The first deep learning method [3] of leveraging the use of CNN's [4] and achieving state-of-the-art results is titled "image super resolution using deep convolutional networks," or commonly known as SRCNN. This architecture consists of three layers, i.e., one layer for patch extraction is then passed into the non-linear mapping layer, which finally is given to the reconstruction layer. The patch layer extracts dense patches from the image. The non-linear layer consists of 1×1 convolutional filters, which changes or increases the number of channels and includes a non-linearity factor. And finally, the reconstruction layer maps the output from the non-linear layer to a high-resolution representation. This gave rise to multiple new architectures such as Very Deep Super Resolution (VDSR) [5], which is 20 layers deeper than SRCNN, Fast Super-Resolution Convolutional Neural Network (FSRCNN), which includes convolutional layers for shrinking, non-linear mapping, and expansion [6] and the Efficient Sub-Pixel CNN (ESPCN) [7] implements sub-pixel shuffle layers, after which upscaling is performed. Similarly, the introduction of a new type of CNN Architecture known as Residual CNNs or ResNets [8] paved the way for residual-based Super-Resolution Architectures such as Enhanced Deep Residual

Networks for Single Image Super-Resolution (EDSR) [9], which utilizes optimization by removing unnecessary residual layers and performs optimized expansion of the model. Wide Activation for Efficient and Accurate Image Super-Resolution (WDSR) [10] implements a low-rank convolution to widen activation.

Following the lines of CNN architecture, we shall take a look at a particular CNN architecture known as the U-Net architecture. This architecture can be understood by splitting it into individual paths, downsampling, and upsampling. The input passes through a series of convolutional layers and is downsized to a suitable representation. This feature map is then passed through a series of upsampling layers to generate an output with the original representation shape. The convolutional layers in the downsampling and upsampling path have skip connections connecting each downsample layer to the corresponding upsample layer where the information is concatenated. This is done to provide transfer information representation from the downsampling layers to the upsampling layers. This architecture is mainly used for image segmentation, where the model is trained with images as input and corresponding object segmentation maps as output. The original paper was proposed for image segmentation in biomedical imaging. In this paper, we have leveraged the U-Net architecture for super resolution to avoid usage of intensive compute architectures by using large pre-trained models and still generate reasonable SR images for the task of segmentation; we use a state-of-the-art architecture known as MR-CNN or Mask R-CNN. This architecture is a combination of both object detection and image segmentation models. The first part of the model is the region proposal network or RPN network, which proposes the object bounding boxes. The second part is the Binary mask classifier, which generates the object segmentation masks for every class. The backbone of the model is a ResNet101 Architecture. Our proposed architecture combines both the U-Net architecture with the MR-CNN to generate high-resolution foreground object segmentation. Foreground/background segmentation is the process of segmenting the foreground objects from the background using segment maps and vice versa.

Some of the applications of our architecture include increasing the quality of foreground images /objects in video conference platforms such as zoom, teams, and other media platforms, replacing the background with another background, filter effects, and video editing. The authors of [11] propose a real-time, high-quality background replacement architecture where the current background of any image can be changed, keeping the existing foreground objects intact. The results are so precise that they preserve the details of the object while merging/blending them with the background. This architecture achieved 30 FPS in 4 k resolution and 60 FPS in HD resolution. They have implemented the segmentation model with Atrous Spatial Pyramid Pooling with DeepLabV3 [12]. The decoder network predicts the Alpha matte of the background, the foreground mask, and the foreground segmented image. While this paper focuses on effortless blending and merging of the foreground with new backgrounds, our architecture focuses on increasing the quality of foreground objects alone while disregarding background during inference.

3 Proposed Architecture

The U-Net Architecture is a CNN architecture that expanded with few changes in the CNN architecture. It was developed to carry out image segmentation on biomedical images where the target is to segment and highlight the area of infection. We leverage this model for the task of super resolution in our experiments. The main motive behind using U-Net architecture is that the model can learn the required and necessary transformations locally for each pixel or region in an image—this is possible due to the presence of the upsampling layers in the model. The downsampling model learns the image’s spatial features, and each corresponding downsampling layer is then connected to its corresponding upsampling layers in the model hierarchy. This allows the combination of different spatial features from different layers, enabling the model to precisely localize more regions of interest. For foreground super resolution, we first take an input image from the dataset and pass it to the MR-CNN model, which gives us the segmented masks for the objects/regions of interest in the image. The masks are mainly Boolean value masks which, when multiplied with the original LR image, gives us the segmented regions from the LR image. This represents the foreground segmented image which is then subtracted from the LR image to get the segmented background of the LR image. We pass the segmented foreground image to the U-Net model, which produces a corresponding SR output of the foreground image, which is then added back to the segmented background image to get the final output. The flow of the proposed architecture is shown in Fig. 1.

4 Implementation and Training

4.1 Dataset and Augmentation

The DIV2K [13] dataset is a collection of high-resolution RGB images compiled for the NTIRE2017 and NTIRE2018 Super-Resolution Challenges. DIV2K dataset stands for DIVerse 2 K Resolution Images. Each of these 1000 images has at least 2 K pixels on one of their axes (vertical or horizontal). The entire dataset has been divided into 800 high-resolution images for training, 100 high-resolution images for validation, and 100 high-resolution images for testing. For conducting the experiments for this paper, we resized these images to a size of 256×256 for the high-resolution images, which are then resized to 256×256 images to half their size and then rescaled to 256×256 , i.e., 50% of the original 256×256 h image. A sample subset of images can be seen in Fig. 2.

We took the original DIV2K training set for the training set, which consists of 800 2 K resolution images. For computation purposes, we resized each of these images to 256×256 resolution. The LR input for these corresponding HR images was generated by rescaling the HR images by a factor of 50 and then resized back to 256

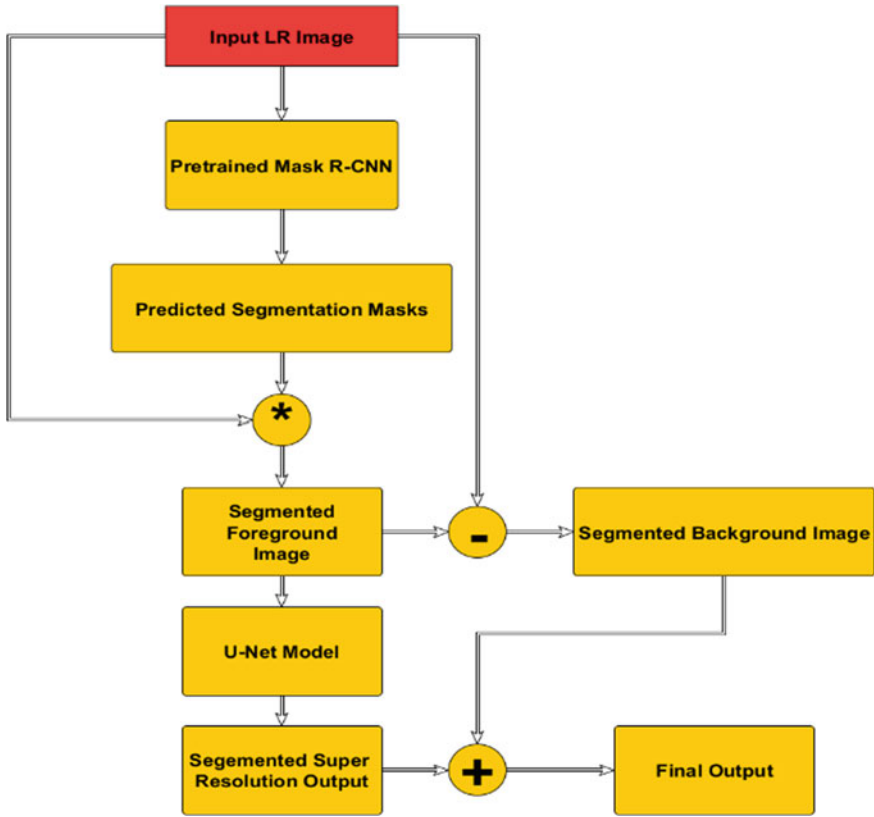


Fig. 1 Proposed architecture flow for foreground super resolution using U-Net and MR-CNN

$\times 256$ size. For faster computation, we created a generator function that randomly selects a batch of 10 images from the dataset and converts it to the required input format. A sample input LR and HR pair can be seen in Fig. 3.

4.2 Model Architecture

For super resolution, we had first trained a vanilla CNN U-Net Model as shown in Fig. 5. The U-Net model consists of two convolutional downscaling blocks followed by two upscaling blocks. The output of each downscaling block is added to the corresponding upscaling block. We add a dropout layer to the first downscaling block to reduce computation. The downscaling and upscaling blocks are shown in Fig. 4. A pre-trained MR-CNN was used for obtaining the segmentation masks. The output



Fig. 2 A subset of images from DIV2K training set

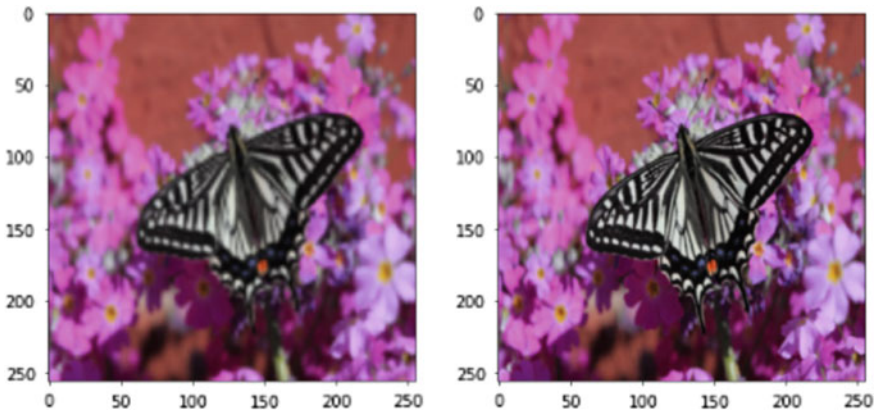


Fig. 3 (Left) 256×256 rescaled LR (Scale: 50) (Right) 256×256 original HR resized

of the MR-CNN on the LR Image is preprocessed to obtain the segmented image, which is then used as input for the U-Net model. A pre-trained implementation of Mask R-CNN [14] was used for foreground segmentation.

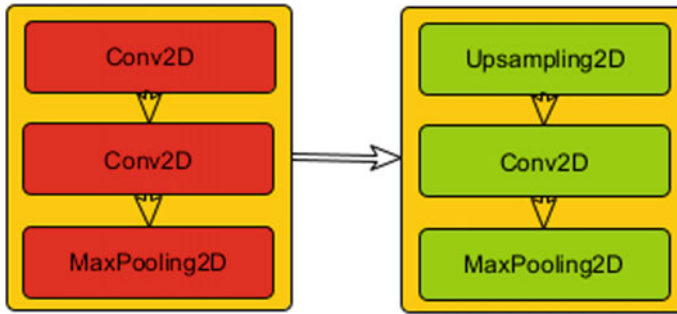


Fig. 4 Left: downscaling block. Right: upscaling block

4.3 Loss Function and Optimizer

The MSE Loss or Mean Squared Error Loss is used for calculating pixel-wise loss between the original HR image and the U-Net output. This is calculated and compared across all three channels in an RGB image. The mean of each pixel difference is calculated and then squared.

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (x_{ij} - y_{ij})^2$$

The optimizer used here is the Adam Optimizer. The advantage of using the Adam optimizer is that the optimizer has adaptive learning rates for each parameter. The optimizer estimates moments (moving average of the parameters) and uses them to optimize a function. It is more robust since it is a combination of RMSprop and SGD with momentum optimizers. Moreover, it is beneficial for models with a large number of parameters or for training on large datasets.

4.4 Training

We trained the vanilla U-Net model for 40 epochs with a step size of 20. The model quickly dropped from a loss of 6402.0 to a final loss of 83.72, as shown in Fig. 6.

Further training can decrease the loss value. The loss value per epoch for every four epochs is shown in Table 1. To prevent the model from stagnating at a particular loss value, we leveraged Tensorflow's ReduceLROnPlateau functionality to modify the learning rate if the model starts to become stationary at a particular loss value/range.

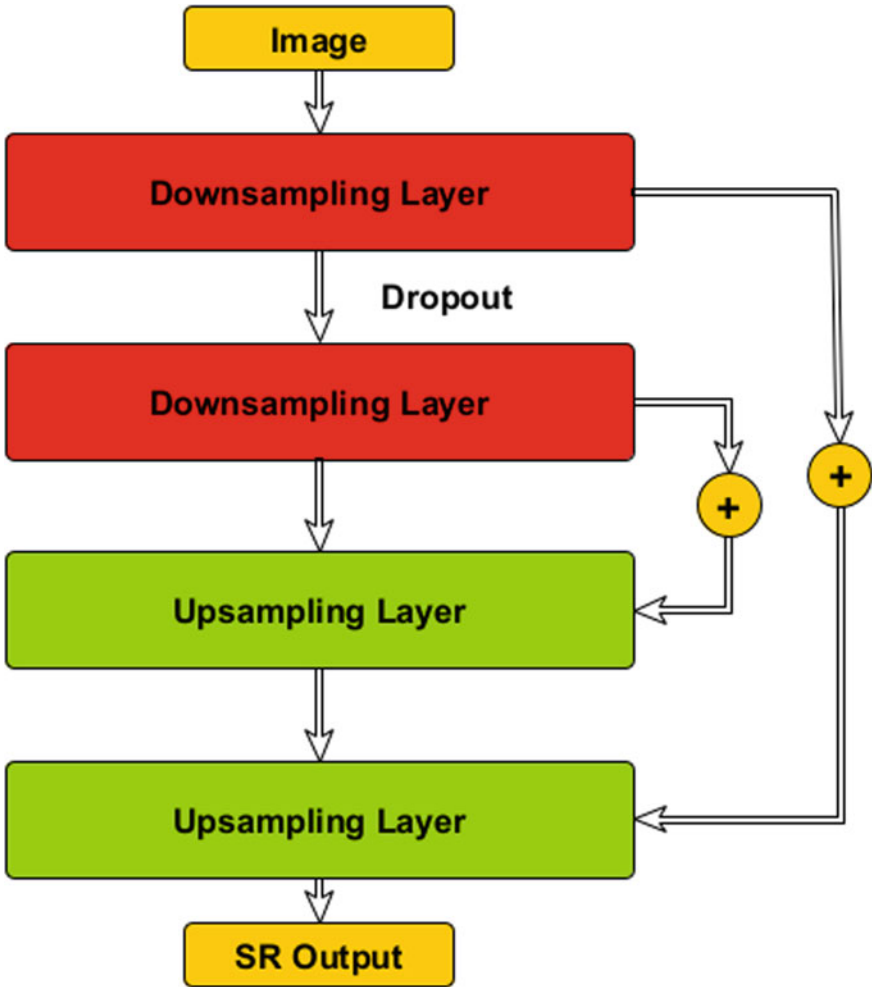


Fig. 5 Model architecture of vanilla CNN U-Net

5 Evaluation Metrics

5.1 PSNR—Peak Signal-To-Noise Ratio

The peak signal-to-noise ratio is a concept that shows the ratio between the power of a signal and the noise that affects its representation. In terms of images, it is the ratio between the largest possible power of an image and the largest power of corrupting noise (image noise such as jittered edges, deformities, etc.) that affects the quality and appearance of an image.

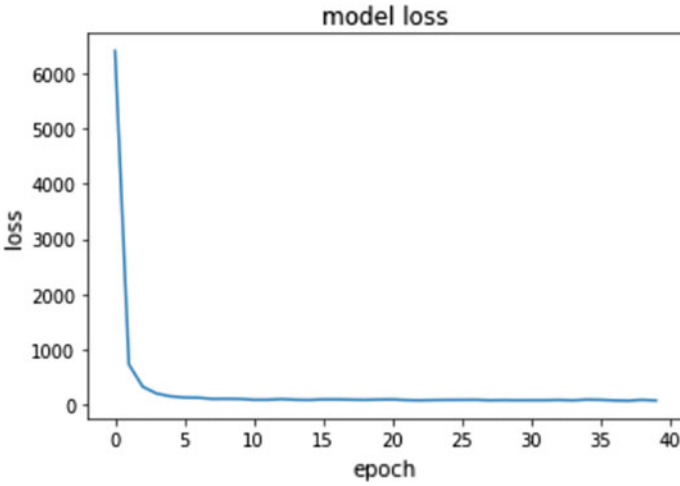


Fig. 6 Training Loss Curve: Training Loss at each epoch for 40 epochs

Table 1 Training loss for every four epochs and final epoch loss

Epoch	Loss
1	6402.04
5	158.56
9	114.28
13	109.07
17	103.53
21	103.12
25	94.47
29	91.10
33	93.99
37	85.24
40	83.72

$$PSNR_{(x,y)} = \frac{10 \log_{10}[\max(\max(x), \max(y))]^2}{|x - y|^2}$$

5.2 SSIM—Structural Similarity Index

The Structural Similarity Index (SSIM) [15] measures the similarity between images. It can be used to evaluate the quality of one image over another. It is an image quality metric that analyzes the significant impact of three characteristics present in an image: luminance, contrast, and structure.

$$SSIM_{(x,y)} = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

5.3 Universal Image Quality Index

The UQI metric [16] is computed by considering the effects of image distortion on the quality of subjective measurements. Experiments have revealed that it performs significantly better compared to the commonly used mean squared error metric.

$$UQI_{(x,y)} = \frac{(4\sigma(xy))\dot{x}\dot{y}}{(\sigma_x^2 + \sigma_y^2)(\dot{x}^2 + \dot{y}^2)}$$

6 Results and Discussion

The U-Net model was evaluated on the validation set, consisting of 100 h Images and their corresponding computed LR images as input. After the inference, the mean increase in SSIM of the validation set was calculated. We had achieved an average increase of about 6.2253 in the SSIM score and about 8.7743 in the PSNR score, and an increase of about 0.4768 in UQI metrics. The overall increase for each metric is shown in graph Fig. 7. The algorithm for metric percentage increase is shown in Algorithm 1. Some samples of the original SR Output are shown in Figs. 8 and 9.

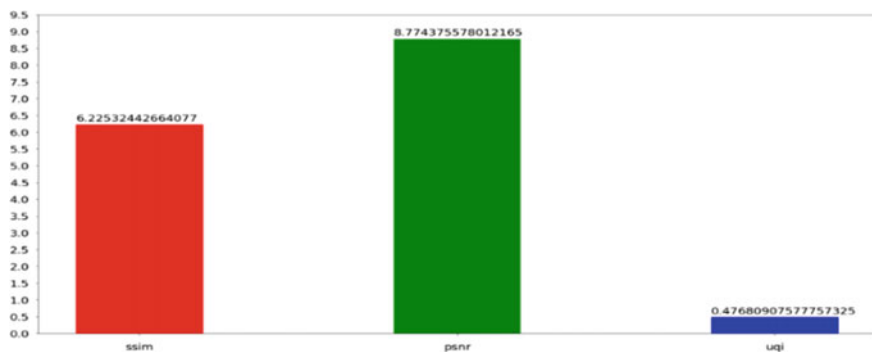


Fig. 7 Evaluation metric increase after super resolution on 50% quality images

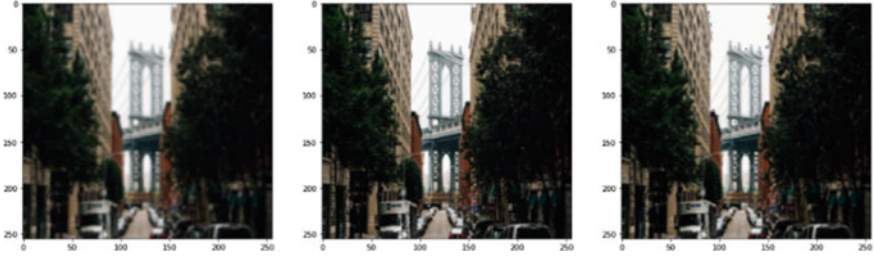


Fig. 8 Left–Right: original LR Image (0.8152 SSIM—25.5518 PSNR), original HR image, predicted SR image (0.8724 SSIM—26.2194 PSNR)

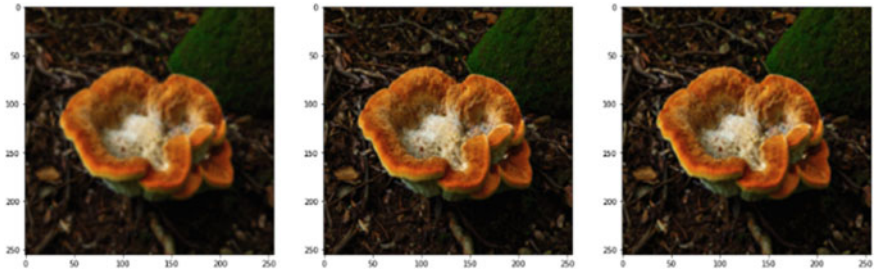


Fig. 9 Left–Right: Original LR Image (0.8499 SSIM—29.0403 PSNR), Original HR Image, Predicted SR Image (0.9099 SSIM—31.2586 PSNR)

```

Data: ValidationImageBatch(X, Y)
Result: AverageIncreaseMetricScore
Overallpercent = 0
N = length(X)
For x in X, y in Y:
    prediction = Model(x) # Make prediction on input image
    orig_score = Metric(x, y) # Calculate metric value
    pred_score = Metric(prediction, y) # Calculate new metric
    score_increase = (pred_score-orig_score)/orig_score*100
    overallpercent+=score_increase
return overallpercent/N
    
```

Algorithm 1. Validation method for a particular metric

We then compared the metrics between the vanilla CNN U-Net for super resolution and our foreground super resolution flow. The LR Images are passed into the MR-CNN, and the masked regions are passed into the U-Net model, and we evaluate the metrics on the combined foreground segmented SR output and LR background image. Two examples are shown for reference with corresponding metric scores in Figs. 10 and 11. In some cases, the output showed an increase in quality. In other images, the quality decreased drastically. Since the focus of our paper is to increase foreground or

object in context resolution, this can be ignored. Here are a few examples with SSIM and PSNR scores and all three metrics before and after foreground super resolution in Figs. 12 and 13.

Image	SSIM	PSNR	UQI
LR and HR	0.9056	29.2197	0.9932
LR and Output	0.8888	29.2082	0.9923

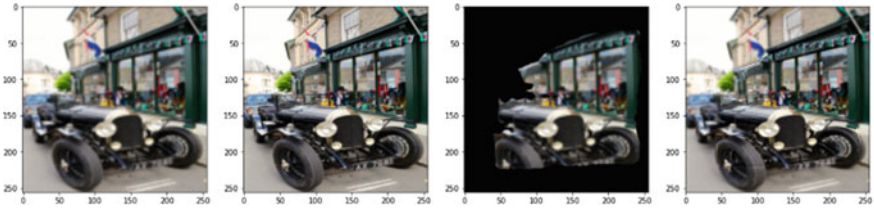


Fig. 10 Left–Right: Original LR Image (0.8716 SSIM—25.1606 PSNR), Original HR Image, Masked LR Image, Masked SR + Background LR (0.8919 SSIM—25.6003 PSNR)

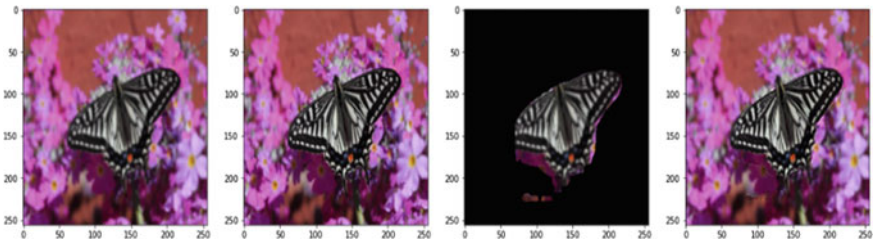


Fig. 11 Left–Right: Original LR Image (0.9306 SSIM—26.1428 PSNR), Masked SR + Background LR (0.9382 SSIM—27.2623 PSNR)



Fig. 12 Left–Right: Original LR Image, Original HR Image, Masked SR + Background LR; Below: Metric values of original LR and foreground SR

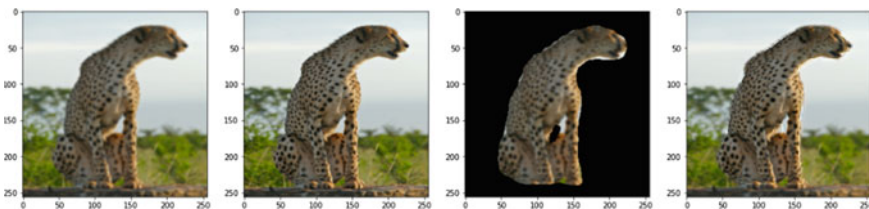


Fig. 13 Left–Right: Original LR Image, Original HR Image, Masked LR Image, Masked SR + Background LR; Below: Metric Values of Original LR and Foreground SR

Image	SSIM	PSNR	UQI
LR and HR	0.9252	29.9149	0.9945
LR and Output	0.9258	27.4436	0.9935

But in some cases, the imposed super resolution mask with the original LR creates unwanted white and jittery boundary edges, as shown in Fig. 14. This might be the leading cause for a decrease in quality. This can be further solved by applying smoothing filters to merge those edges with the background.



Fig. 14 Jittered Edge between background LR and masked SR region

7 Conclusion and Future Work

Using the DIV2K dataset, the application of foreground image super resolution was explored using a Vanilla CNN U-Net as the SR Model and MR-CNN for foreground segmentation. The U-Net trained was able to increase the quality of the LR Images quickly. For the foreground or portrait images, the metric values showed a slight increase in SSIM value because the model specifically increases only the quality of the object in context (segmented object). This was possible because even if the background of the image is increased in quality, Due to blurred effect in portrait, there is not much difference between the background of an LR image and an HR image. Thus the metric increase for the background does not matter. But the PSNR metric showed a drastic decrease due to jittered white edges (borders) when superimposing the SR Segmented Image back to the LR image. These edges add to the noise factor of the image, and thus we see a decrease in PSNR values. Further developments can be made to negate these blurred edges, such as using filtered blurring or merging edges with the neighboring pixels or the background. Moreover, we have only implemented a simple U-Net architecture for our experiments due to constrained CPU and GPU resources. By implementing state-of-the-art architectures such as WDSR, EDSR, etc., we can achieve 2 K–4 K resolution quality for the foreground images. These improvements combined with the base architecture can be applied to various applications such as the custom creation of portrait images, superimposing of images onto various backgrounds similar to the video filters provided by video conferencing platforms such as Zoom, Microsoft Meetings, and other social media platforms such as Snapchat and Instagram.

Acknowledgements We thank Ydealogy Ventures for providing us with the required computational resources. The company's resources and their constant feedback and insights while working on this paper were highly valuable.

References

1. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. CoRR
2. He K, Gkioxari G, Doll P, Girshick RB (2017) Mask R-CNN. CoRR
3. Dong C, Loy CC, He K, Tang X (2015) Image super-resolution using deep convolutional networks. CoRR
4. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional networks. Neural Inform Process Syst
5. Kim J, Kwon Lee J, Mu Lee K (2016) Accurate image super-resolution using very deep convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1646–1654
6. Dong C, Loy CC, Tang X (2016) Accelerating the super-resolution convolutional neural network. Proceedings of the European conference on computer vision

7. Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. CoRR
8. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. CoRR
9. Lim B, Son S, Kim H, Nah S, Mu Lee K (2017) Enhanced deep residual networks for single image super-resolution
10. Yu J, Fan Y, Yang J, Xu N, Wang Z, Wang X (2018) Thomas huang: wide activation for efficient and accurate image super-resolution. CoRR
11. Lin S, Ryabtsev A, Sengupta S, Curless B, Seitz S, a Kemelmacher-Shlizerman I (2020) Real-time high-resolution background matting. CoRR
12. Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking Atrous convolution for semantic image segmentation. CoRR
13. Agustsson E, Timofte R (2017) NTIRE 2017 challenge on single image super-resolution: dataset and study. The IEEE conference on computer vision and pattern recognition (CVPR) workshops, 1122–1131
14. Abdulla W (2017) Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow, Github
15. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process*, 600–612
16. Wang Z, Bovik AC (2002) A universal image quality index. *IEEE Signal Process Lett* 9(3):81–84

SMS Spam Classification Using PSO-C4.5



D. Saraswathi and D. Sowmya

Abstract In this modern era, with the increased use of mobile communication, spam messages have also become common. SMS spam was sent for commercial benefits thus distributing spam links and unnecessary messages. Even now, few are seen as the victim of Fraudulent SMS. The proposed study identifies spam in SMS. The proposed method identified optimized features using a particle swarm optimization algorithm and classified them using a decision tree algorithm C4.5. This combination of PSO-C4.5 produced better performance when compared with the conventional algorithms.

Keywords SMS spam · Classification · Machine learning · Decision tree · PSO

1 Introduction

The SMS concept was developed in the year 1984 in the Franco-German Cooperation by Friedhelm Hillebrand and Bernard Ghillebaert and was evolved since then, revolutionizing the communication sector. Reports suggest that over 2.71 billion people own a smartphone in the year 2019 and about two-thirds of the world population is connected by the cellular network [1]. This surge in the usage of mobile phones directly/indirectly turned SMS services into a multi-million-dollar commercial industry [1].

The unsolicited messages or junk messages sent in bulk that target a mobile phone are referred to as SMS spam which poses a major problem over recent years [2, 3]. This paper proposes to focus on one such algorithm used in spam message detection and to give an efficient method to achieve the same [4].

D. Saraswathi (✉) · D. Sowmya
Department of Computer Science, PSG College of Arts & Science, Coimbatore, India
e-mail: saraswathi.ds@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
R. J. Kannan et al. (eds.), *Computer Vision and Machine Intelligence Paradigms for SDGs*, Lecture Notes in Electrical Engineering 967,
https://doi.org/10.1007/978-981-19-7169-3_4

41

2 Problem Statement

The major problem with SMS spams is storage issues, wasting users' time, which also leads to other types of threats like money fraud indulgences, virus attacks, and message disclosure. Thereby, detecting spam SMS leads to a binary text classification problem where the text messages are categorized into either spam or ham messages. Thus, there becomes a necessity for the mobile industry to find a suitable and efficient algorithm to filter spam in the best way possible.

3 Research Objective

The ultimate objective of the paper is to classify ham or spam messages from the SMS dataset and propose an efficient algorithm for the same. The proposed method in this paper identifies important patterns to detect SMS spam by using PSO and classified using decision tree C4.5.

4 Review of Literature

This section discusses the literature survey, made to identify the commonly used text pre-processing techniques [5, 6], feature extraction [7], feature selection [8], and classifiers [9] for SMS spam detection, and conclusions were drawn from the respective survey [10].

4.1 *Review of Text-Processing*

The process of text pre-processing helps to remove the inconsistency in the dataset and enables us to produce better results in the later process of Spam detection [11, 12]. The papers [13, 14] encouraged the use of tokenization, removal of stop words, and punctuations to the SMS dataset before proceeding with it to the feature extraction process. The papers [15–17] used pre-processing techniques like conversion to lower case, removal of stop words, and stemming were performed.

4.2 *Review on Feature Extraction*

On a spam detection system, the next step is feature extraction. Based on the features used, it helps to check whether a message is a ham or spam [14, 18]. Some of the

features were identified in the existing systems [19, 20] such as Spam keywords, lowercased and uppercased words, URLs, mobile number, message length, and special symbols were extracted to classify the SMS [16, 21].

4.3 Review on Feature Selection

Feature selection is an important step in classification and is used to select a small subset of features from an original set of features. Since it reduces irrelevant features by improving classification accuracy, it becomes more essential when databases are growing in size and complexity. The selection process is expected to bring benefits in terms of better-performing models, computational efficiency, and simpler more understandable models, one such is the PSO algorithm, used to select the optimized features. PSO-C4.5 was produced with better accuracy than C4.5 [22]. Also, few literature surveys discussed hybrid method produced better performance than conventional classification [23].

4.4 Review on Classifiers

Classifier plays a vital role in classifying ham or spam messages [24]; Comparative study of different classifiers, namely, K nearest neighbor, Naïve Bayes, and Decision tree with the establishment of various machine learning algorithms for SMS spam detection was made [25, 26]. Compared with various algorithms and experimental results show that Decision Tree had got a good accuracy rate than the other algorithms [9, 24]. From the understandings made in the survey, it was seen that C4.5 decision tree algorithm is commonly used for classification problems, thus C4.5 was selected to classify SMS spam messages.

5 Research Contribution

With the insights obtained in Sect. 4, this section puts them together. The proposed work comprises various phases for SMS spam classification. The phases are SMS repository, SMS Chooser, feature extraction, SMS feature selection using PSO, normalization, and classification. The proposed study has depicted in Fig. 1

Initially, a collection of SMS was stored in an SMS repository to check its authenticity of the SMS. SMS chooser is used to choosing SMS from the SMS Repository. After that, each SMS is tokenized. This tokenize process reveals a list of keywords and links. Feature extraction is used to identify important patterns present in the SMS that can be stored and processed.

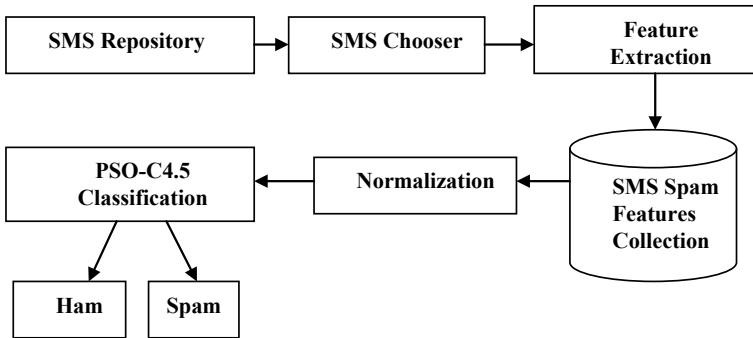


Fig. 1 Proposed architecture

Feature selection is the optimization technique that reduces the number of input variables to improve the performance of SMS spam classification. According to a literature survey, particle swarm optimization (PSO) has provided the best performance when compared with other algorithms. The proposed system selected PSO for an optimization approach that is used for feature selection tasks. After selecting an optimized feature using PSO, a model has been constructed using the C4.5 decision tree algorithm and validated.

From the insights drawn from the existing system, the decision tree C4.5 algorithm was chosen for SMS classification as it can handle missing attribute values and avoids overfitting of data. The hierarchical decomposition of the training data in decision tree C4.5 classifiers helps to learn the decision rules to classify the SMS. A hierarchical tree is generated by using different attributes and their values. The split ratio and gain ratio is calculated by using the list of SMS attributes. Information gain is calculated for the target attributes. The one with the highest gain ratio is taken as the root node for the tree. The decision nodes are labeled with different split ratio values. These attributes in the decision node have low gain ratio, compared to the root node. This procedure is repeated until all the features are classified as spam or spam SMS. The leaf node has the least Information gain and comprises the final class label of the message as ham or spam. SMS classification is based on the attribute-based split. Finally, classification results have been submitted to the user interface.

6 Data Collection and Data Sampling

The SMS for the present study were collected from the kaggle.com dataset [27], which is a publically available collection of SMS. This SMS collection is labeled as “spam” and “ham” by the evaluator. 5,574 SMS have been collected out of which 4,826 messages are ham messages and 748 messages as spam as per the dataset taken for the study.

7 Experimental Results

To evaluate the performance of machine learning classifiers, a confusion matrix is used [7]. The metrics of this matrix represent the various measures applied to the test sample. The structure of a confusion matrix [4, 6] for a two-class problem is presented in Table 1, with two different classes namely positive occurrence and negative occurrence. The rows of a confusion matrix indicate the actual class whereas the columns indicate the predicted class. The researchers [19, 20] from the study made in Sect. 4 used four parameters of the confusion matrix to assess the performance of the algorithms. These attributes are True Positive (TP) Rate, True Negative (TN) Rate, False Positive (FP) Rate, False Negative (FN) Rate, Accuracy Rate, and Error Rate [21].

Where

TP = Number of correctly classified ham messages as ham

TN = Number of correctly classified spam messages as spam

FP = Number of incorrectly classified spam messages as ham

FN = Number of incorrectly classified ham messages as spam

From the survey made on classifiers, it was seen that machine learning algorithms like SVM, NB, DT (C4.5), and KNN were commonly used for Spam message classification. In Table 2, it is seen that among the classifiers taken, the Decision Tree (DT) C4.5 classifier outperforms all other classifiers. The experimental results have been shown that the C4.5 decision tree classifier does not handle irrelevant features in the dataset that may reduce the performance of the accuracy rate.

The present study has been selected the PSO algorithm for optimized feature selection and the C4.5 Decision Tree for classification. The proposed PSO + C4.5 technique has been compared with the C4.5 decision tree classifier, PSO + C4.5 approach outperforms the conventional C4.5 decision tree classifier. The PSO + C4.5 approach has proven that optimized feature classification generates better results when compared to the un-optimized decision tree classification. The experimental results of the PSO + C4.5 approach are shown in Table 3 and Fig. 2.

Table 1 Structure of confusion matrix

Total no. of instances		Predicted class	
		Ham	Spam
Actual class	Ham	TP	FN
	Spam	FP	TN

Table 2 Comparative study of various classifiers

Algorithm	Accuracy rate (%)
KNN	86.04
SVM	86.24
DT	89.07
NB	87.93

Table 3 Accuracy Rate of C4.5 VS PSO + C4.5

Classification algorithm	Accuracy rate (%)	Error rate(%)
DT C4.5	89.07	10.93
DT C4.5 + PSO	95.58	4.42

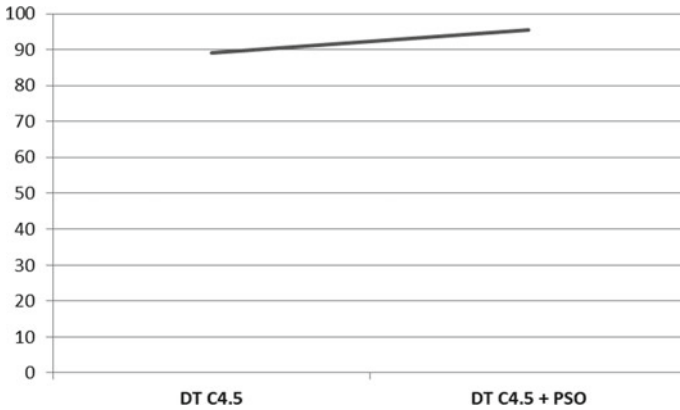


Fig. 2 Experimental results of Decision tree C4.5 with the hybrid approach

8 Conclusion and Future Enhancement

The experimental results reveal that the PSO + C4.5 technique was found to exhibit better performance when compared to the conventional C4.5. The combination of the PSO + C4.5 technique was also found to classify the SMS spam more effectively, thus validating the rationale of the study. The approach developed and experimented tested in the present study offers plenty of scope for future research. A combination of the classification algorithm and other optimization-based techniques could be used to develop more intelligent approaches to tackle the challenges in the domain.

References

1. Rubin Julis M, Alagesan S (2020) Spam detection In Sms using machine learning through text mining. *Int J Scient Technol Res* 9(2)
2. Ntoulas A, Najork M, Manasse M, Fetterly D (2010) Detecting spam web pages through content analysis. 15th International world wide web conference committee, Published by Association for Computing Machinery, Inc
3. Radhakrishnan A, Vaidhehi V (2017) Email classification using machine learning algorithms. *Int J Eng Technol*, 335–340
4. Shirani-Mehr H (2013) SMS spam detection, using machine learning approach, Stanford University, CSS029: Machine Learning projects
5. Ruskanda FZ (2019) Study on the effect of pre-processing methods for spam email detection. *Indonesian J Comput* 4(1):109–118. ISSN: 2460-9056

6. Gupta P, Dubey RK, Mishra S (2019) Detecting spam Emails/SMS using naive Bayes and support vector machine. *Int J Scient Technol Res* 8(11)
7. Mussa DJ, Jameel NGM (2019) Relevant SMS spam feature selection using wrapper approach and XGBoost Algorithm. *Kurdistan J Appl Res (KJAR)* 4(2). Print-ISSN: 2411–7684|Electronic-ISSN: 2411–7706
8. El Bakrawy LM (2019) Hybrid particle swarm optimization and Pegasos algorithm for spam email detection. *Adv Syst Sci Appl* 19(3)
9. Charbuty B, Abdulazeez A (2021) Classification based on decision tree algorithm for machine learning. *J Appl Sci Technol Trends* 2(1):20–28
10. Egele M, Kolbitsch C, Platzer C (2011) Removing web spam links from search engine results. *J Comput Virol*, 51–62
11. Angiani G, Ferrari L, Fontanini T, Fornacciari P, Iotti E, Magliani F, Manicard S (2016) A comparison between preprocessing techniques for sentiment analysis in Twitter , KDweb, Italy
12. Fattahi J, Mejri M (2021) SpaML: a bimodal ensemble learning spam detector based on NLP techniques. *IEEE 5th international conference on cryptography, security and privacy*
13. Saidani N, Adi K, Allili M (2017) A supervised approach for spam detection using text-based semantic representation. Springer International Publishing
14. Oluwatoyin O, Bodunde A, Titus G, Ganiyu A (2019) An improved machine learning-based short message service spam detection system. *I. J. Comput Netw Inform Secur*, 40–48
15. Hameed SM (2021) Differential evolution detection models for SMS spam. *Int J Electr Comput Eng* 11(1)
16. Gupta SD, Saha S, Kumar S (2021) SMS spam detection using machine learning. *J Phys Conf Ser* 1797:012017
17. Prieto VM, Alvarez M, Lopez-Garcia R, Cacheda F (2012) Analysis and detection of web spam by means of web content. *International proceedings of the 5th information retrieval conference*
18. Jameel NGM (2018) SMS spam detection using association rule mining based on SMS structural features. *J Theoret Appl Inform Technol*. ISSN: 1992-8645, E-ISSN: 1817–3195
19. Verma RK, Gupta S, Saini Y, Libang (2020) Content-based SMS spam detection. Delhi Technological University
20. Alqahtani SS, Alghazzawi D (2019) A survey of emerging techniques in detecting SMS spam. *Trans Mach Learn Artif Intell* 7(5)
21. Minhaz Hossain SMD, Aashiq Kamal KMD, Sen A, Sarker IH (2021) TF-IDF feature-based spam filtering of mobile SMS using machine learning approach. *Inform Technol Data Manag*. <https://doi.org/10.20944/preprints202109.0251.v1>
22. Primartha R, Tamab BA, Arliansyaha A, Miraswan KJ (2019) Decision tree combined with PSO-based feature selection for sentiment analysis. *J Phys Conf Ser* 1196
23. Palanisamy C, Kumaresan T, Varalakshmi SE (2016) Combined techniques for detecting email spam using negative selection and particle swarm optimization. *Int J Adv Res*
24. Jun LG, Nazir S, Khan HU, Ul Haq A (2020) Spam detection approach for secure mobile message communication using machine learning algorithms. *Mach Learn Appl Cryptogr*
25. Safar N, Kalavathi K (2015) Performance comparison between Naïve Bayes, Decision Tree, and K-Nearest Neighbour. *Int J Emerg Res Manag Technol* 4(6). ISSN: 2278-9359
26. Selvapattu S, Patel PV (2020) SMS spam detection. *Int J Innov Sci Eng Technol* 7(6)
27. www.kaggle.com

Automated Sorting, Grading of Fruits Based on Internal and External Quality Assessment Using HSI, Deep CNN



P. Rahul Ganesh , R. Priyatharshini, M. Sarath Kumar, and A. Raj Kumar

Abstract Good quality fruits demand is expanding due to the ascent in the crowd. Gross domestic product of the multitudinous nations depends upon its export export plays a significant role in GDP. After harvesting they're washed, sorted, graded, pressed, and put down. Out of every one of these stages grading and sorting of fruits are vital way. The main end is to plan an automated system that improves the standard, upgrades the creation productivity, decreases the work cost of the fashion, and assesses the internal quality of the fruits. As per the agricultural and food products export development authority pomegranates, mangoes, bananas, papayas, and orange account for the larger portion of fruits exported from our country. Effective discovery is achieved using hyperspectral imaging, CNN frame. Superior performance than treating, sorting, and grading grounded on redundant classes in the CNN frame through the proposed architecture.

Keywords Internal quality · Hyperspectral Imaging (HSI) · CNN framework

1 Introduction

India's different environment guarantees accessibility to all blends, fruits, and vegetables. It positions as second vegetable and fruit product creation on the planet, after China. Orange and apple are more exported in quantities compared to other fruits in the existing system used to detect only one kind of fruit based on the image and through edge detection method and only external defects can be identified, a technical limitation of the systems. The accuracy of the existing system is very low due to its less efficient approach and the existing systems only look for the external quality of the fruit, therefore missing to detect the internal quality leading to export of defected fruit. The proposed system enables to recognize and grade multiple fruits in the same system, and it can assess internal quality, it is done through the use of deep CNN model. The accuracy of the high due to the use of a efficient approach. It

P. Rahul Ganesh (✉) · R. Priyatharshini · M. Sarath Kumar · A. Raj Kumar
Easwari Engineering College, Chennai, India
e-mail: reonrahul8@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
R. J. Kannan et al. (eds.), *Computer Vision and Machine Intelligence Paradigms for SDGs*, Lecture Notes in Electrical Engineering 967,
https://doi.org/10.1007/978-981-19-7169-3_5

can be implemented in real time and eliminates the need of heavy external hardware as it only uses camera. As it automates the process, it saves time and the need for manual intervention. The computational cost of the overall project and the implementation cost is highly affordable. It has very low maintenance cost. With deep CNN model creation with adjusted layers utilized for training a distinctive number of image dataset, the test results have high accuracy in classification.

2 Related Works

In [1] the inward nature of nectarines the yellow flesh called as big top and white flesh called Malique has been assessed utilizing hyperspectral imaging. As hyperspectral pictures of flawless undamaged fruits were obtained in the spectral range. The discovery of disorder and classification as indicated by a set up firmness limit were performed utilizing PLS-DA. The expectation of the Internal Quality Index (IQI) identified with ripeness was done utilizing PLS-R. The main factors were chosen utilizing interval-PLS.

In [2] the purpose of this study behind the systemization of fruit sorting utilizing image processing is to make greater fruit sorting, quality support, and creation and to reduce labor numbers. It is a need for an automated system that brisk, and quality feature identification just as the snappy apportioning of fruits is completed. The system proposes a total start to finish system computerization dependent color, surface component extractions, and utilizing image segmentation methods to portion the defected region and characterize the nature of the fruit.

In [3] the individual method is illustrated by thinking about ghostly/picture procurement mode followed by data extraction from obtained information. At that point, possibilities of various chemometric, hyperspectral data reduction, and image processing and highlight extraction strategies utilized in both procedures have been embodied for anticipating the imperfections, fecal tainting, and dry matter. The spectral signature of the picture is exceptional for that material as it varies because of its physical and chemical properties.

In [4] a capable and successful machine vision system that uses deep learning methodologies and group procedures to build a cost-effective and non-destructive solution for computerizing the visual assessment of the freshness and appearance of fruits. To choose the ideal model for fruit grading, we trained, tested, and thought about the presentation of many profound models like as ResNet, MobileNetV2, EfficientNet, NASNet, and DenseNet. With a camera and a microcontroller, the proposed device also provides continuous visual investigation with minimal effort. The real-time system extracts multiple instances of natural compounds from an image and then grades each fruit accurately.

3 Overview of Proposed Idea

The proposed system proposes a total start to finish structure of computerized and has a highly efficient grading and sorting based on Internal and External quality assessment using CNN where we train image-based datasets for external quality and hyperspectral images of the fruits for internal quality. The proposed system in which image acquisition takes place loading the external images of the fruits and is preprocessed (resized). Then for internal quality assessment, in which image acquisition takes place loading the hyperspectral images of the fruits and is preprocessed (resized, enhanced), [5, 6]. Then the images with defects are segmented (HSV, complemented, holes filled) and fed into the CNN layers which identify the quality (Fig. 1).

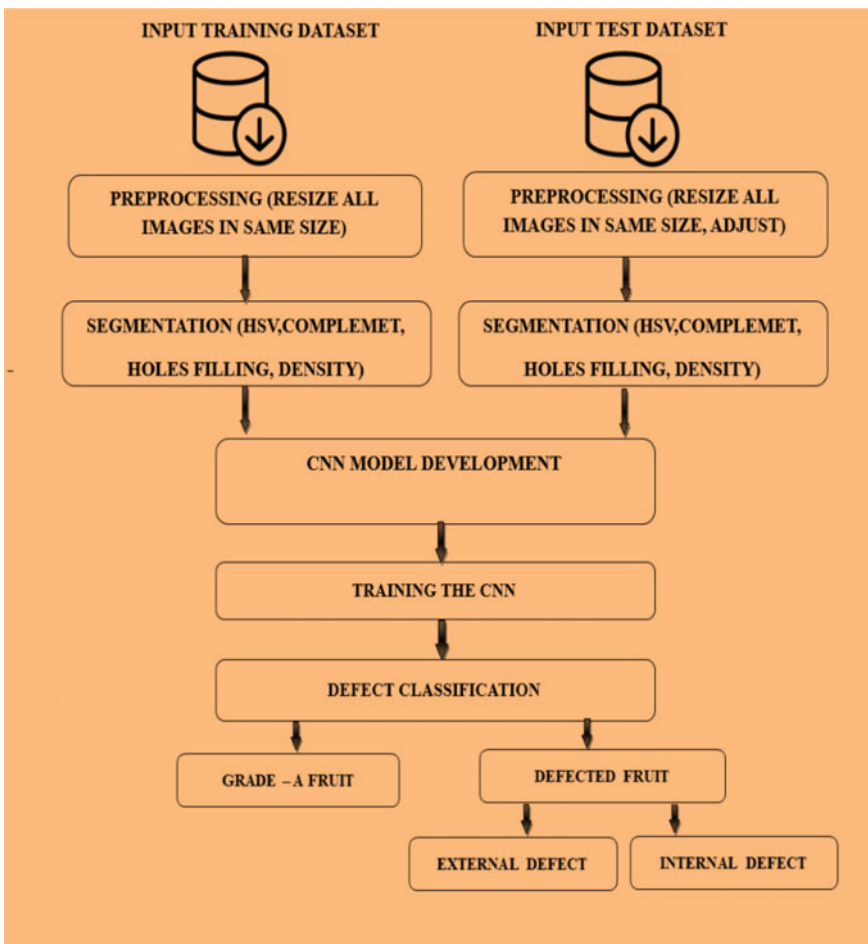


Fig. 1 Proposed system architecture

Fig. 2 Segmented image

3.1 Preprocessing

The fruit image data is acquired from ImageNet and the hyperspectral image dataset of the fruit is separated and divided into training and test dataset. The first step is to download images into MATLAB. Read and display an input image. Using the `imread` command, the acquired picture is also fed into code. It is the operation of entering (acquiring) a picture from a source, primarily from a hardware source, for processing in image processing. It is the first step in the workflow order because, without an image it is impossible to process. The image that is acquired is completely untouched. In large image dataset deep learning can learn features automatically. All the images of the datasets are resized into same size (standard CNN input image size is 200×200 pixels in dimension), the pre-processing step.

3.2 Segmentation

For meaningful and accurate analysis dataset can require techniques. This process aids in the extraction of significant visual properties, which may then be used to interpret information. Data segmentation strategies include finding, deleting, and replacing poor or missing data, as well as segmenting the region of interest. The segmentation includes morphological operations, regional properties, and the operations that are performed to obtain the segmented region or image are RGB to HSV, black and white conversion, complement, holes filling, removing small objects, and density selection (Fig. 2).

3.3 CNN Model Development

CNN is a widely used deep learning algorithm that is used to take the input image which is further classified. The CNN model contains three types of layers, such as pooling layers, convolutional layers, and fully connected layers. CNN has many convolutional layers which are exchanged. The last layer is the fully connected layer before which pooling layers are present. The result of each layer is based on the

commitment of the going with layer, [7–9]. Width and height are the elements of the images. The profundity is the amount of info color channels. Now, for deep learning you should give the training data to the classifier so it can fabricate a model (Fig. 3).

Pseudocode:

```
layers = [imageInputLayer([230 230 3])  
convolution2dLayer (5,512,'Padding',2,'stride',1)  
crossChannelNormalizationLayer(5)  
reluLayer  
maxPooling2dLayer(3,'Stride',2,'Padding',0)  
fullyConnectedLayer(4)  
softmaxLayer  
classification
```

Experimental results are shown below:

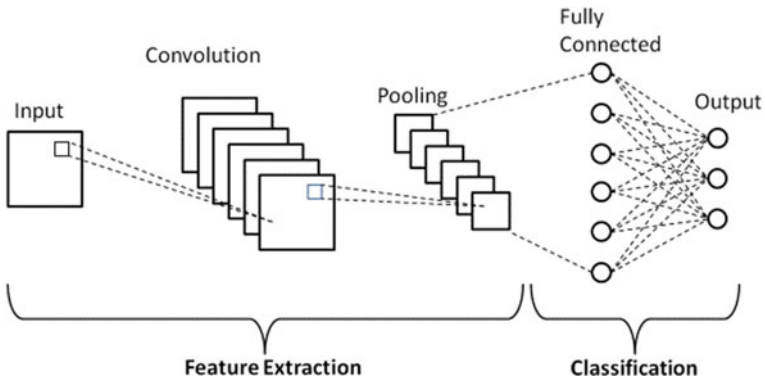
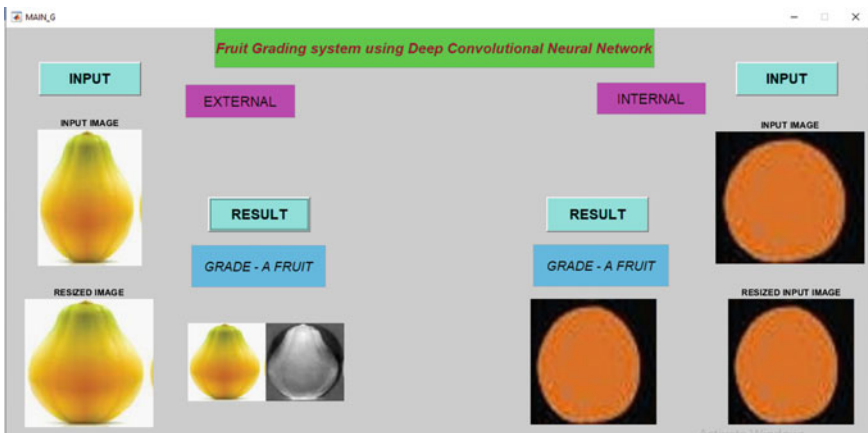
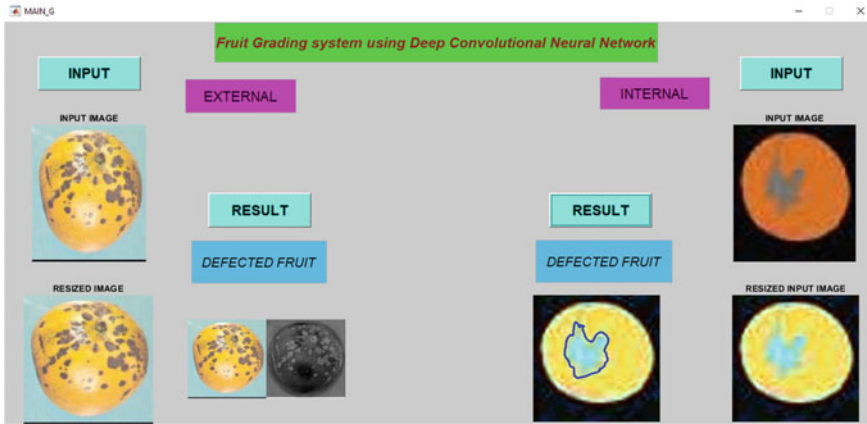


Fig. 3 CNN model



4 Experimental Results and Discussion

4.1 Experimental Setup

Unlike other methods which consider only outside appearance to detect the fruits' qualities but our system also considers the internal quality of the fruit with astonishing accuracy of over 98.57%. The dataset represents the internal and external qualities of the fruit (Fig. 4).

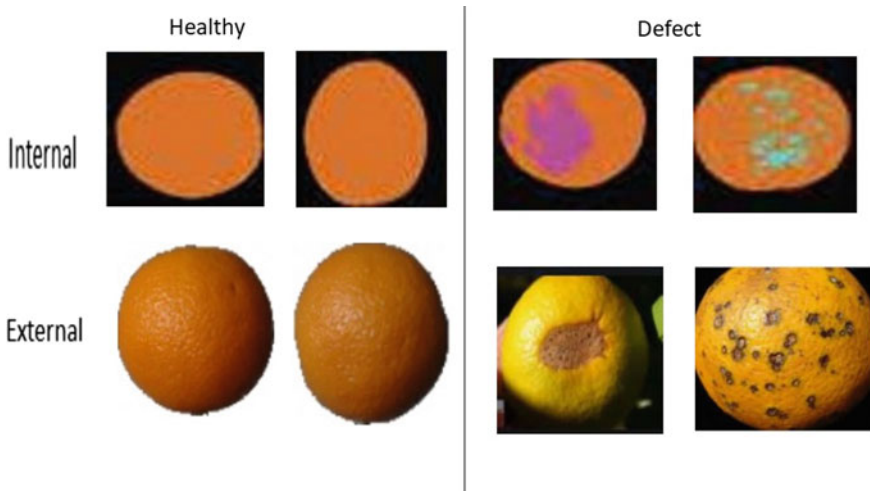


Fig. 4 Sample of Internal and external qualities of fruit

- 50 Internal (Hyperspectral) images of fruits.
- 100 External images of fruits.

4.2 Performance Measures

Accuracy is a measure that describes how well the model performs across all classes. It is beneficial when all classes are of equal importance. It's calculated by dividing the number of right guesses by the total number of predictions. Machines learn by using a loss function. It is a method of determining how effectively an explicit algorithm models the data, [10]. It's a strategy for assessing how well explicit the algorithm models the given data. In the event that predictions deviate a lot from actual output loss function would hack up an extremely huge number. Loss function figures out how to decrease the error in prediction (Fig. 5 and Table 1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

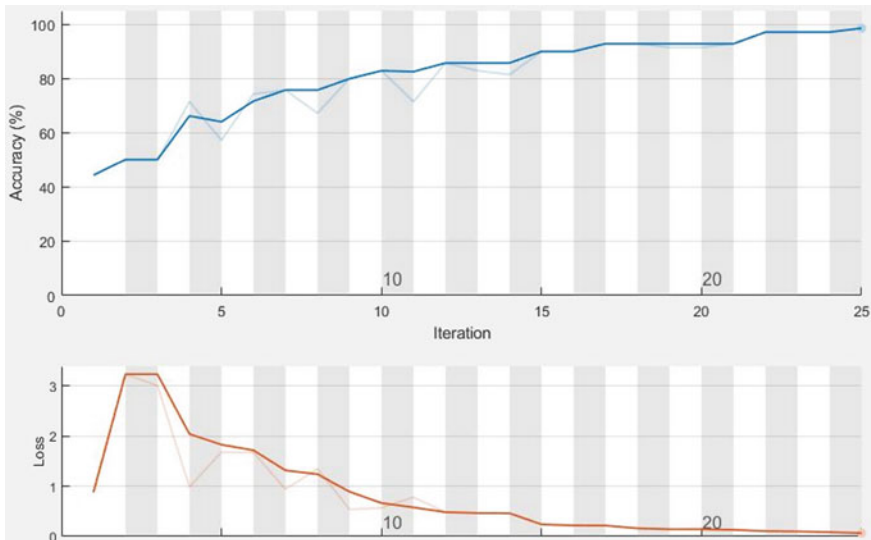
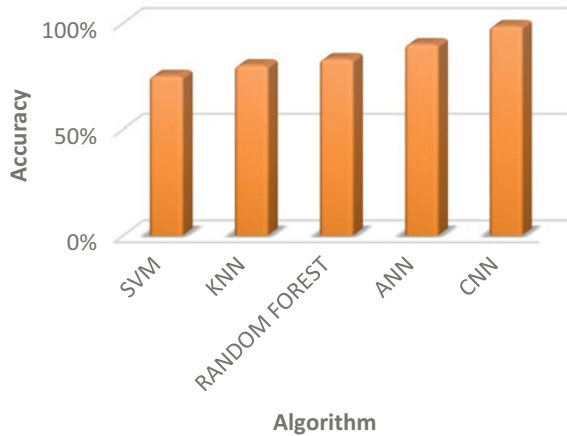


Fig. 5 Accuracy and loss graph

Table 1 Comparative performance analysis

Methods	Accuracy (%)
SVM	75
KNN	80
Random forest	83
ANN	90
CNN	98.57

Fig. 6 Comparative performance analysis

4.3 Experimental Results

The motivation behind why Convolutional Neural Networks (CNNs) show improvement over other neural organizations on images and videos is that the convolutional layers exploit the inherent properties of images. Convolutions, simple feedforward neural networks like feedforward network don't perceive any request in their information sources, [11–13]. If you rearranged every one of your images similarly, the neural network would have the same performance it has when prepared on not rearranged pictures (Fig. 6).

5 Conclusion

In conclusion, there are many papers published which considers only the external appearance of the fruit but in our proposed system fully automated and effective sorting and grading system based on both internal and external quality assessment using CNN which yields higher accuracy and better sorting, grading compared to existing systems. The proposed system enables to recognize and grade multiple fruits in the same system and also it can assess internal quality, it is done through the use of

the deep CNN model. The accuracy of the high due to the use of an efficient approach. Furthermore, most of these systems have attempted to distinguish between matured and young fruits, despite the fact that the number of young fruits is more important for forecasting long-term production fluctuations.

References

1. Munera S, Blasco J, Amigo JM, Cubero S, Talens P, Aleixos N (2019) Use of hyperspectral transmittance imaging to evaluate the internal quality of nectarines. *Biosys Eng* 182:54–64. <https://doi.org/10.1016/j.biosystemseng.2019.04.001>
2. Abbas HMT, Shakoor U, Khan MJ, Ahmed M, Khurshid K (2019) Automated sorting and grading of agricultural products based on image processing. In: 2019 8th international conference on information and communication technologies (ICICT), pp 78–81. IEEE. <https://doi.org/10.1109/ICICT47744.2019.9001971>
3. Chandrasekaran I, Panigrahi SS, Ravikanth L, Singh CB (2019) Potential of near-infrared (NIR) spectroscopy and hyperspectral imaging for quality and safety assessment of fruits: an overview. *Food Anal Methods* 12(11):2438–2458. <https://doi.org/10.1007/s12161-019-01609-1>
4. Ismail N, Malik OA (2021) real-time visual inspection system for grading fruits using computer vision and deep learning techniques. *Inf Process Agric* (2021) <https://doi.org/10.1016/j.inpa.2021.01.005>
5. Tang Y, Gao S, Zhuang J, Hou C, He Y, ChuX, ... Luo S (2020) Apple bruise grading using piecewise nonlinear curve fitting for hyperspectral imaging data. *IEEE Access* 8:147494–147506. <https://doi.org/10.1109/ACCESS.2020.3015808>
6. Liu Q, Wei K, Xiao H, Tu S, Sun K, Sun Y, Tu K (2019) Near-infrared hyperspectral imaging rapidly detects the decay of postharvest strawberry based on water-soluble sugar analysis. *Food Anal Methods* 12(4):936–946. <https://doi.org/10.1007/s12161-018-01430-2>
7. Zhang L, Gui G, Khattak AM, Wang M, Gao W, Jia J (2019) Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot. *IEEE Access* 7:56028–56038. <https://doi.org/10.1109/ACCESS.2019.2899940>
8. Pan X, Sun L, Li Y, Che W, Ji Y, Li J, ... Xu Y (2019) Non-destructive classification of apple bruising time based on visible and near-infrared hyperspectral imaging. *J Sci Food Agric* 99(4):1709–1718. <https://doi.org/10.1002/jsfa.9360>
9. Zhuang J, Hou C, Tang Y, He Y, Guo Q, Miao A, ... Luo S (2019) Assessment of external properties for identifying banana fruit maturity stages using optical imaging techniques. *Sensors* 19(13):2910. <https://doi.org/10.3390/s19132910>
10. Ji Y, Sun L, Li Y, Li J, Liu S, Xie X, Xu Y (2019) Non-destructive classification of defective potatoes based on hyperspectral imaging and support vector machine. *Infrared Phys Technol* 99:71–79. <https://doi.org/10.1016/j.infrared.2019.04.007>
11. Li J, Zhang R, Li J, Wang Z, Zhang H, Zhan B, Jiang Y (2019) Detection of early decayed oranges based on multispectral principal component image combining both bi-dimensional empirical mode decomposition and watershed segmentation method. *Postharvest Biol Technol* 158:110986. <https://doi.org/10.1016/j.postharvbio.2019.110986>
12. Beyaz (2018) Harvest glove and LabView based mechanical damage determination on apples. *Scientia Horti* 228:4955. <https://doi.org/10.1016/j.scienta.2017.09.049>
13. Tan W, Sun L, Yang F, Che W, Ye D, Zhang D, Zou B (2018) Study on bruising degree classification of apples using hyperspectral imaging and GS-SVM. *Optik* 154:581–592. <https://doi.org/10.1016/j.ijleo.2017.10.090>

Pest Detection Using Improvised YOLO Architecture



M. Sujaritha , M. Kavitha , and S. Roobini 

Abstract The speedy and reliable classification of plant disease/pest is essential to preventing productivity loss and loss or diminished quantity of agricultural commodities. Machine learning methodology can be used to obtain the solution. Deep learning has achieved significant advancement in the development of image processing in modern years, greatly outperforming previous approaches. Researchers are very interested in understanding how to apply deep learning to swot plant and pests detection. Deep learning, which is extremely popular in image processing, has offered many innovative precision farming applications in recent decades. In this investigation, deep learning models are adapted to the task at hand using transfer learning and deep feature extraction approaches. The given work takes into account the used pre-trained deep models for feature extraction and fine-tuning RCNN (Region with Convolution Neural Network) and YOLO (You Only Look Once) are used to classify the features extracted by deep feature extraction. Improvised YOLO is used which has proven pest prediction of about 95%. The performance of current research is compared, and common datasets are introduced. This paper examines potential obstacles in real-world applications of deep learning-based plant disease and pest detection. Data from genuine infection and pest pictures is used in the investigations. For performance evaluation, the accuracy is computed and compared.

Keywords YOLO · CNN · RCNN · Pest

M. Sujaritha (✉) · M. Kavitha
Sri Krishna College of Engineering and Technology, Coimbatore, India
e-mail: sujaritham@skcet.ac.in

M. Kavitha
e-mail: kavitham@skcet.ac.in

S. Roobini
SNS College of Technology, coimbatore, India
e-mail: srruby13@gmail.com

1 Introduction

Human civilization now has the potential to generate enough food to feed more than 7 billion people thanks to modern innovations. Nevertheless, a numeral of factors such as climate change [1], pollinator reject, and plant diseases [2] continue to pose a threat to food security. Pests and pathogens are a worldwide danger to foodstuff security, but they might also be destructive for smallholder farmers whose livelihoods are dependent on healthy crops. Small-scale farmers contribute more than 80% of agricultural production in the on the rise countries and reports of yield losses are not less than 60% due to pests and infections are widespread [3].

Workers do passive observing in many circumstances as they go about their on a daily basis tasks. The downside to this approach is that by the time the plague is recognized, a substantial proportion of harm has already been done. In big farms, early pest detection required a more organized methodology. Traps are, without a doubt, the most commonly used technique for pest monitoring [4, 5]. The great majority of research in the literature is concerned with the second stage. The first step is generally only handled straightforwardly: an explanation of how the data was obtained is frequently included. The third phases are largely outside the purview of research.

2 Literature Review

Gutierrez et al. [6], who conducted a comparative analysis using a combination of pre-trained deep learning model as a mixture of models implemented with machine learning and computer vision, stimulated the current study. The main goal of the [6] study is to improve pest identification accuracy by using current frameworks like TensorFlow and Keras to construct a deep convolutional neural network (CNN). In addition, several recent pre-trained models may be applied to the dataset to assess accuracy. Table 1 depicts the overall survey of pest management and algorithms used and their accuracy.

2.1 Pest Detection Methods

The goal of uncovering methods is to separate a confident target bug from the rest of the scene in a picture. This corresponds to a dual classification using the classifications “target visible” and “target missing.” K-means clustering is a vector quantization approach for grouping a set of comments into k-clusters or k-classes. The image is first divided into 100×100 blocks by the algorithm. The RGB and L*a*b* color spaces are then utilized as the foundation for an algorithm that pre-selects probable cluster centers before using K-means clustering to categorize each pixel. Using ellipse eccentricity rules, erroneous objects are removed.

Table 1 Comparison of algorithms with accuracy based on the problem

References	Problem	Pest	Input	Classifier used	Accuracy
Barbedo et al. [4]	Detection	Psyllids	Part of Image	Squeezenet CNNs	0.69–0.92
Espinoza et al. [7]	Detection	Wester flower, whiteflies	Part of original image	Multilayer ANN	0.90–0.95
Dawei et al. [8]	classification	10 species	Pre-prososed image or resized image	AlexNet (CNN)	0.94
Deng et al. [9]	Classification	10 Species	NNSC, SIFT, LCP Features	SVM	0.84
Dimililer et al. [10]	Classification	8 Species	Part of Image	ANN	0.92
Liu et al. [11]	Classification	16 Species	Image	CNN + RPN + PSSM	0.745
Wang et al. [12]	Detection	3 Species	Image	CNN + DecisionNet	0.62–0.91
Wang et al. [13]	Detection	Whiteflies	image	K-means cluster	0.92–0.98
Batool et al. [14]	detection	Different species	Processed image	K-nearest neighbours algorithm	0.79–0.89
Yoa et al. [15]	Detection	Several	Original images	Normalized cuts, watershed, k-means	0.956
Xia D et al. [16]	Classification	24 species	Re-sized image	VGG19	0.89
Ebrahimi et al. [17]	Detection	Thrips	HIS color channels	SVM	0.98
Limiao Deng [18]	Detection	Different species	Cropped images	LCP + SVM	0.85
Metwalli et al. [19]	classification	Food images	Pre-posed	DenseNet	0.83
Kumar et al. [20]	Classification	Different images of urban waste products	Pre-processed	YOLO v3	0.85

2.2 Pest Classification Methods

The difficulty of classifying pests is significant since a classification like this must not only distinguish among the embattled species but also contract with nontargeted species, which might be many. The closest coldness between the retrieved attribute

vector and the reference vectors associated with each class was used to classify each item as a whitefly, aphid, or thrip. Xia [16] utilized the watershed method to partition the insects, then used the Mahalanobis distance to extract color characteristics from the YCrCb colour space. For the classification of eight pest species. Dawei [8] used transfer learning to classify 10 species in pictures collected in the field using a pre-trained AlexNet CNN. Metwalli [19] present the DenseFood model, which is a densely linked CNN model with several convolutional layers. The phrase “You Only Look Once” is abbreviated as YOLO.

To identify objects, the technique just takes a single forward propagation through a neural network, as the name indicates. This indicates that a single algorithm run is used to forecast the whole picture. The CNN is used to anticipate multiple bounding boxes and class probabilities at the same time. There are several variations of the YOLO algorithm. Tiny YOLO and YOLOv3 are two popular examples.

3 YOLO V3 Architecture

YOLO because of its velocity and accuracy, this algorithm is very fashionable. YOLOv3’s network design is made up of three distinct networks. The first is Darknet-53, which serves as the network’s backbone. The detecting layers, also known as YOLO-layers, come next, followed by an upsampling network. Figure 2 depicts the network structure. The backbone network, Darknet-53, is utilized to extract features from the input picture. The basic components of Darknet-53 are residual blocks and 53 convolutional layers. A residual block is made up of two 3×3 and 1×1 convolutional layers linked together via a shortcut connection. Figure 3 depicts the Darknet-53 architecture in its entirety.

Figure 1 shows the overview of YOLOv3 structure. The numbers below each layer show the dimension decrease of the input at that layer. The gray layer is the input layer. The blue layers are part of the backbone network, Darknet-53. The red layers are upsampling layers and the yellow layers are YOLO-layers.

4 Improved YOLO V3 Architecture

The model divides the images into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence (C) for those boxes and class probabilities (CP). The predictions are encoded as an $S \times S \times X (B * C + CP)$ Tensors. Dataset: Identifying a species from a photograph is a difficult task. The categorization of a picture is based on the assumption that the image contains just one species. However, in general, we want to identify ALL of the species in a photograph. Thankfully, biologists and taxonomists have created a taxonomic hierarchy to classify and organize species. Insects, spiders, crustaceans, centipedes, millipedes, and other arthropods are included in the ArTaxOr data set. Figure 2 depicts the overall working of improvised YOLO.

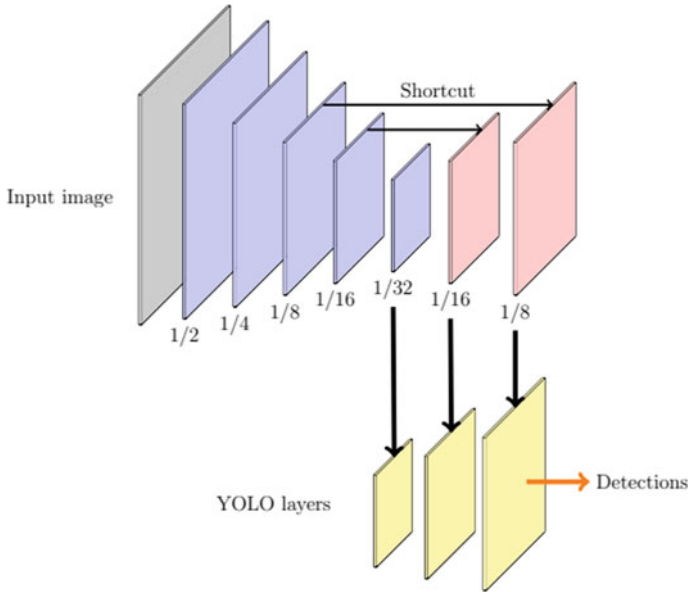


Fig. 1 Overview of YOLO V3

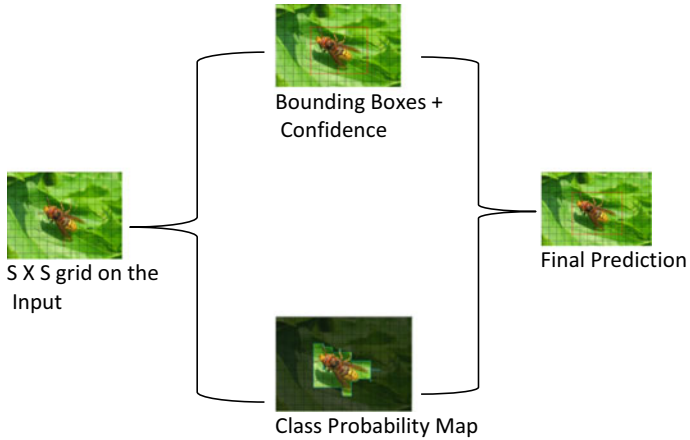


Fig. 2 Improved YOLO V3 architecture

The dataset consists of images of arthropods in jpeg format Araneae (spiders), adults, juveniles, Coleoptera (beetles), adults, Diptera (true flies, including mosquitoes, midges, crane fly, etc.), adults, Hemiptera (true bugs, including aphids, cicadas, planthoppers, shield bugs, etc.), adults and nymphs, Hymenoptera (ants, bees, wasps), adults, Lepidoptera (butterflies, moths), adults, Odonata (dragonflies, damselflies), adults, Orthoptera (grasshoppers, locusts, crickets, etc.)

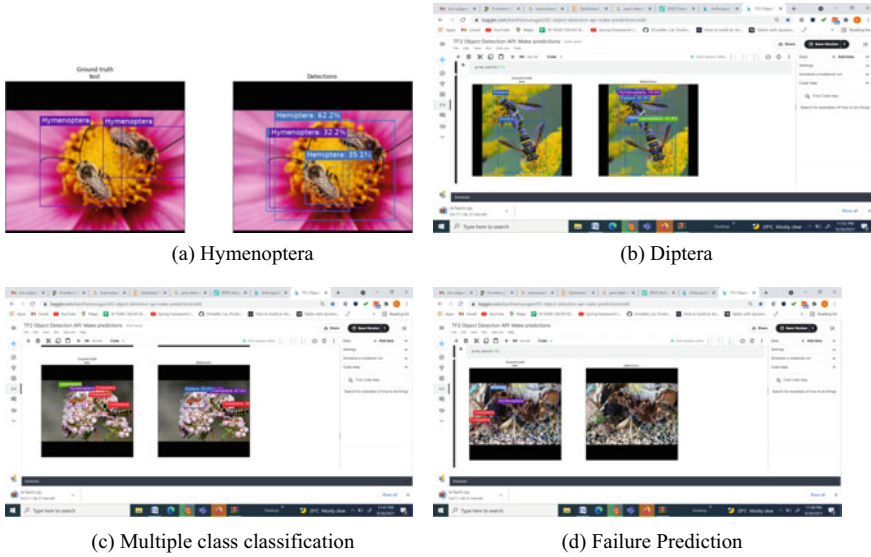


Fig. 3 Sample predictions of different class

Figure 3 predicts the pest in the picture. Accuracy of the pest prediction is also marked in the image. Ground truth image and predicted image are specified for comparing the accuracy of the prediction. The model fails to predict the class is also projected in the figure. Table 2 shows the accuracy comparison of each class.

5 Results and Discussions

See Fig. 3 and Table 2.

6 Conclusion

It is difficult to automate pest monitoring. Machine learning algorithms have evolved to the point where the apparatus desirable to develop a precise system with real-world application is now readily obtainable. Congregation data that is reflective of the enormous variety observed in live-out is difficult, more common, and procedures to permit consumer research get more refined, this may become less of an issue in the vicinity of future. but, as mentioned all through this paper, there are unmovng numerous explore gaps to be filled, implying that pest monitor mechanization will remain a fascinating study topic for numerous years. As proposed YOLO V3 architecture shows around 95% of accuracy in different pest predictions. Comparatively

Table 2 Comparing the accuracy of different classes

Algorithm	Prediction accuracy classes								Overall Accuracy	mAP	Time/ms
	Araneae	Coleoptera	Diptera	Hemiptera	Hymenoptera	Lepidoptera	Odonata	Orthoptera			
RCNN	95.21	84.32	93.49	87.81	84.06	71.08	81.29	78.08	84.41	95.74	35.69
Yolo	95.65	89.97	92.78	89.56	85.62	89.98	89.56	93.88	95.875	96.99	29.21

YOLO v3 works better and provide good result than RCNN. Adding more images for training will help to reduce the failure cases. In case if we have less images we recommend to use image argumentation for better and more accuracy.

References

1. Tai AP, Martin MV, Heald CL (2014) Threat to future global food security from climate change and ozone air pollution. *Nat Clim Chang* 4:817–821. <https://doi.org/10.1038/nclimate2317>
2. Strange RN, Scott PR (2005) Plant disease: a threat to global food security. *Phytopathology* 43:83–116. <https://doi.org/10.1146/annurev.phyto.43.113004.133839>
3. Harvey CA, Rakotobe ZL, Rao NS, Dave R, Razafimahatratra H, Rabarijohn RH et al (2014) Extreme vulnerability of smallholder farmers to agricultural risks and climate change in madagascar. *Philos Trans R Soc Lond B Biol Sci* 369:20130089. <https://doi.org/10.1098/rstb.2013.008>
4. Barbedo JGA, Castro GB (2019) Influence of image quality on the identification of psyllids using convolutional neural networks. *Biosyst Eng* 182:151–158
5. Yen AL, Madge DG, Berry NA, Yen JDL (2013) Evaluating the effectiveness of five sampling methods for detection of the tomato potato psyllid, *Bactericera cockerelli* (Sulc) (Hemiptera: Psylloidea: Trioziidae). *Aust J Entomol* 52:168–174
6. Gutierrez A, Ansuategi A, Susperregi L, Tubío C, Ranki ć I, Lenčič L (2019) A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. *J Sens*
7. Espinoza K, Valera DL, Torres JA, López A, Molina-Aiz FD (2016) Combination of image processing and artificial neural networks as a novel approach for the identification of *Bemisia tabaci* and *Frankliniella occidentalis* on sticky traps in greenhouse agriculture. *Comput Electron Agric* 127:495–505
8. Dawei W, Limiao D, Jiangong N, Jiyue G, Hongfei Z, Zhongzhi H (2019) Recognition pest by image-based transfer learning. *J Sci Food Agric* 99:4524–4531
9. Deng L, Wang Y, Han Z, Yu R (2018) Research on insect pest image detection and recognition based on bio-inspired methods. *Biosyst Eng* 169:139–148
10. Dimililer K, Zarrouk S (2017) ICSPI: intelligent classification system of pest insects based on image processing and neural arbitration. *Appl Eng Agric* 33:453–460
11. Liu L, Wang R, Xie C, Yang P, Wang F, Sudirman S, Liu W (2019) PestNet: an end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access* 7:45301–45312
12. Wang F, Wang R, Xie C, Yang P, Liu L (2020) Fusing multi-scale context-aware information representation for automatic in-field pest detection and recognition. *Comput Electron Agric* 169:105222
13. Wang Z, Wang K, Liu Z, Wang X, Pan S (2018) A Cognitive Vision Method for Insect Pest Image Segmentation. *IFAC Pap Online* 51:85–89
14. Batool A, Hyder SB, Rahim A, Waheed N, Asghar MA et al (2020) Classification and identification of tomato leaf disease using deep neural network. 2020 international conference on engineering and emerging technologies (ICEET), IEEE, pp 1–6
15. Yao Q, Lv J, Liu QJ, Diao GQ, Yang BJ, Chen HM, Tang J (2012) An insect imaging system to automate rice light-trap pest identification. *J Integr Agric* 11:978–985
16. Xia D, Chen P, Wang B, Zhang J, Xie C (2018) Insect detection and classification based on an improved convolutional neural network. *Sensors* 18:4169
17. Ebrahimi M, Khoshtaghaza M, Minaei S, Jamshidi B (2017) Vision-based pest detection based on SVM classification method. *Comput Electron Agric* 137:52–58
18. Deng L, Wang Y, Han Z, Yu R (2018) Research on insect pest image detection and recognition based on bio-inspired methods. *BioSyst Eng* (Elsevier) 169:139–148

19. Metwalli, Shen W, Wu CQ (2020) Food image recognition based on densely connected convolutional neural networks. 2020 international conference on artificial intelligence in information and communication (ICAIIC), pp 027–032
20. Kumar S, Yadav D, Gupta H, Verma OP, Ansari IA, Ahn CW (2021) A Novel YOLOv3 AlgorithmBased deep learning approach for waste segregation: towards smart waste management. Electronics 10:14. <https://doi.org/10.3390/electronics10010014>

Classification of Fungi Effected Psidium Guajava Leaves Using ML and DL Techniques



Sukanya S. Gaikwad, Shivanand S. Rumma, and Mallikarjun Hangarge

Abstract The paper aims to recognize and identify two types of fungi affected leaves, insect eaten leaf and healthy leaves of Guava (*Psidium Guajava*) plant, as most of the crops are lost to fungi. The aim is to work on the prime crop of our H&K (Hyderabad & Karnataka) region, which will help the farmers of this region as there is no work done on this crop. The dataset is possessed from a fruit-farm of H&K region of Karnataka, India. We have implemented both Machine learning (ML) and Deep learning (DL) techniques to get good results on the collected real time environment dataset. We eliminate the preprocessing step to reduce the time complexity. The performance measures obtained from the experiments shows that the Deep learning model performed well as compared to the ML techniques.

Keywords ML · DL · KNN · LDA · SVM · AlexNet · *Psidium Guajava* · Fungi affected

1 Introduction

India is an agrarian country where agriculture is the primary source of income for the majority of the population. Temperature, weather, soil, and terrain are all unique to us. As a result, the production of fruits and vegetables varies greatly across the country. Fruits and vegetables are produced in orchards, gardens, or along highways where the seed may be carried by the wind, birds, or animals. After mango, banana, and citrus, guava is India's fourth most popular fruit. Guava's scientific name is *Psidium guajava*, which is borrowed from Latin. Maharashtra is India's leading producer of guava.

S. S. Gaikwad (✉) · S. S. Rumma
Department of Computer Science, Gulbarga University, Kalaburagi, Karnataka, India
e-mail: gsukanya116@gmail.com

M. Hangarge
Department of Computer Science, Karnatak Arts, Science, and Commerce College, Bidar,
Karnataka, India

The guava plant may be a host for fungi, algae, and bacteria found in the atmosphere, which can harm the plant's leaves, stems, barks, and fruit. Infection of the guava plant is caused by pathogens such as viruses, bacteria, insects, and fungi, as well as unfavorable environmental circumstances. Among these diseases, half of the yield is lost to fungi. The disease symptoms and the area of the affected leaf determine the type of disease. To recognize these diseases there needs to be continuous and frequent monitoring of the plants by plant pathologists or farmers, which is a tiresome and expensive method as they need to travel far from places to the farm and sometimes it may lead to wrong assumptions of the diseases. Hence we try to develop a fast, accurate, automatic, affordable and easily available technology. This is possible through Machine learning, Image processing, deep learning, Pattern recognition and Computer vision technologies which are research attractions in the current scenario.

There are readily available public datasets on online sites like Kaggle, Plant pathology and at Git-hub repository. These datasets have simple and plain background and many researchers have attained highest accuracy around 95–99%. But, for a real scenario, the real time environment images should be studied [1]. Hence our motto in this paper is to capture the diseased and healthy leaves of Guava plant and classify them according to their respective categories using Deep learning techniques.

We focus on two types of fungi affected Guava leaves, healthy leaf and insect eaten leaf of Guava. The real time environment dataset [2] is possessed from a fruit-farm of H&K region of Karnataka, India. For collecting the fungi affected leaf images of Guava plant, the work is been assisted and guided by Dr. Pooja Suryavanshi from Dept. of Botany, Karnatak Arts Science and Commerce college, Bidar, Karanataka. Moreover, in the overview of dataset section, the symptoms of fungi affected leaf diseases visible through naked eyes are given clearly.

Our contributions in this paper are:

- We have collected our own brand new dataset, as there is no work done on the *Psidium guajava* plant from our H&K region.
- Our goal is to identify 2 different fungi affected, one healthy leaf and insect eaten leaf of guava plant using machine learning classifiers and a CNN model such as AlexNet.
- We propose a model where manual calculating of the features is eliminated, which time is consuming.
- The features are extracted from Deep Learning (DL) model which reduces the time complexity.
- These extracted features are used for classification in both ML and DL models.
- The performance measures like classification accuracy, precision, recall and F1 score are utilized to measure the accomplishment of the ML & DL models.
- Agriculture is the soul of our country, and our efforts will aim to boost productivity by detecting diseases early.

The remainder of the paper is organised as follows: Sect. 2 describes the literature survey, Sect. 3 provides an overview of the database, Sect. 4 describes the method or model used, Sect. 5 summarises the experimental results, Sect. 6 provides an analysis of the models, and Sect. 7 summarises the conclusions and future work.

2 Literature Survey

Zhang et al. [3] have collected the leaves of maize plants from different sites like google and plant village. They worked for 9 different class labels and used GoogLeNet and Cifar 10 model for classification. They got a good recognition accuracy of 98.9% using Cifar 10 model and 98.8% using GoogLeNet model.

Mohanty et al. [4] have worked on plant village datasets for 38 different classes. They used AlexNet and GoogLeNet models and obtained an accuracy of 99.34% using GoogLeNet model and 85.53% using AlexNet model.

Durmuş et al. [1] used the tomato dataset from plant village of 10 different categories. They got a good recognition accuracy of 94.3% using SqueezeNet model and 95.6% using AlexNet model.

Arivazhagan et al. [5] used the customized Convolutional Neural Network (CNN) for classification of own collected leaves of Mango plant. The dataset has 6 different class labels and got a good recognition accuracy of 96.67%.

Xiaoxiao et al. [6] collected their own dataset of tea leaves of 7 different class labels. They used SVM, BP and CNN model for classifying the leaves. And for the collected dataset they obtained a good recognition accuracy of 89.36% using SVM, 87.69% using BP and 93.75% using CNN model.

Howlader et al. [7] worked on Deep CNN model for classifying the Guava leaves. Total of four different class labels were used. The model gave a good recognition accuracy of 98.74%.

Gaikwad et al. [8] developed their own customized CNN model for identifying and classifying the fungi affected leaves of Apple plant. The dataset is collected from Plant pathology. Using the CNN model they got a good recognition accuracy of 88.9%.

Sardogan et al. [9] used the tomato dataset, which has five different class labels. They used the CNN with LVQ model for classification of the dataset. The model gave a good result of 86% using CNN with LVQ algorithm.

Sholihati et al. [2] collected their own dataset of potato plant and also collected images from google and plant village datasets. The dataset has five different categories for classification. Vision geometry group (VGG) 16 and VGG 19 are used for identification of the dataset. Using the CNN models they got a good recognition accuracy of 90% using VGG 19 and 91% using VGG 16.

Vijayakumar [10] used the plant village of tomato plant for 7 different class labels. CNN models like AlexNet and VGG 16 are used for classification. They obtained a good recognition accuracy of 97.29% using VGG 16 and 97.49% using AlexNet model.

This summarizes the classification of leaf diseases using CNN architectures. And the author's from [11–15] have worked on different datasets and used different CNN models for classifying the infected leaves of different plants. The infection is due to bacteria, algae, insect or fungi. We have focused only on fungi affected leaves of *Psidium guajava*, which no author has addressed. And no author has worked on the hybrid approach of using both Deep learning models and Machine learning classifiers. Using this method the time and space complexity of calculating and storing of the features is eliminated.

The authors have used the preprocessed images where background of the images is removed. We have eliminated this step of preprocessing to know till how much extent the background affects the decision of the classifier/model.

3 Overview of Database

A well organized database is needed to measure the achievement of the ML and DL models. The brand new database is introduced, which we have possessed from a fruit- farm of H&K region of Karnataka, India. The collected database comprises of 2 types of fungi affected leaf diseases, insect eaten leaf and the healthy leaf of Guava. 2 variety of mobile phones and 1 camera are utilized to take the images. 3966 images are utilized for recognizing and class labeling the taken images. These taken images are categorized into 4 variety of categories described below.

1. **Leaf spot:** This disease is produced by *Pseudocercospora psidii* fungi. It is seen as small dark brown edges around the leaf. Total of 727 images of leaf spot images are possessed from *Psidium Gujava* plant.
2. **Rust:** This disease is produced by *Puccinia psidii* fungi. It is seen as orange to red pustules on the surface of the leaves. Total 860 images of rust leaf images are taken from *Psidium Gujava* plant.
3. **Insect eaten:** It is caused by insect-eating of the leaves. Mostly it appears around the edges of leaves. Total 323 images of insect eaten leaf images are taken guava plant.

Below figure shows the images of healthy and diseased leaves of the Guava from Figs. 1, 2, 3 and 4.

The summary of our new database is shown in Table 1.

There is no proper dataset available of fungi affected leaf diseases of *Psidium Gujava* plant. Especially in H&K region of Karnataka, there is no work done at all on the said plant. Hence there is no proper benchmarked dataset available; due to the unavailability, we have collected our own new dataset of fungi affected leaves.

Fig. 1 Healthy leaf



Fig. 2 Insect eaten



Fig. 3 Leaf spot



Fig. 4 Rust**Table 1** Summary of database

Class label	Total	Training	Testing
Healthy_leaf	2056	1645	413
Insect eaten	323	257	68
Leaf spot	727	581	144
Rust	860	687	171
Total	3966	3170	796

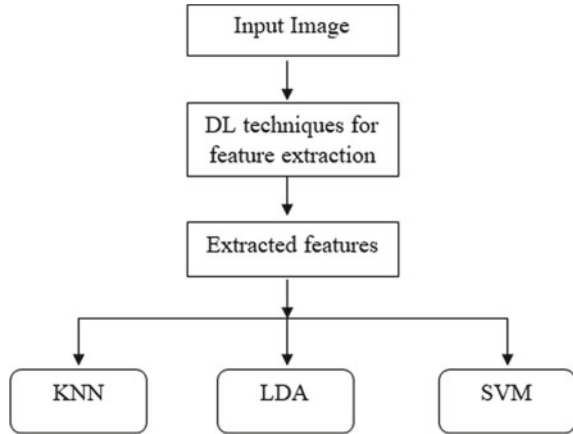
4 Proposed Method/Model

With traditional ML techniques, we have to manually extract the features and select only those features which are prominent to identify the particular class among the calculated features. Manually calculating the features of 1000's of images is space consuming and time consuming. Hence to minimize the time and space requirement, this phase is eliminated in deep learning techniques.

In this paper we try to come up with a thought of extracting features from deep learning models. And then these extracted features are given for machine learning classifiers to classify the images. We also eliminate the preprocessing step of the images. Here we will use three machine learning classifiers which are Support Vector Machine (SVM), Linear Discriminant Analysis (LDA) and K Nearest Neighbour (KNN). Figure 5 shows the hybrid approach of ML and DL models for classification.

We are using the images as it is from the collected real time environment which contains some of the background noise. Because of which the accuracy of the models is reduced a bit. But if we pre-process the image and remove the background then both ML and DL models gives better results.

Fig. 5 Hybrid ML and DL model



4.1 Classification Using Deep Learning Techniques

The function of human brains, which have neurons and synapses that connect them, are inspired by Artificial Neural Networks (ANN). Their key distinguishing feature is their ability to learn under supervision i.e., supervised learning. Convolution Neural Networks (CNN) are an evolution of traditional machine learning algorithms, where they receive inputs (raw data), pass through layers for feature extraction which happens automatically and then outputs the results. The CNN consists of a basic layer such as Convolution layer, Activation function, Max Pooling and Fully Connected layer. Figure 6 shows the CNN architecture.

We use the basic CNN model AlexNet [16] for classification. The model and their training and testing process is implemented on Windows 10, 64 bit operating system containing i7core processor with 12 GB RAM. Below Table 2 shows the details of the execution domain.

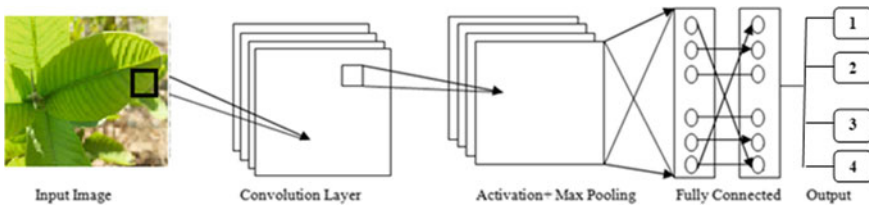


Fig. 6 CNN architecture

Table 2 Execution domain

Name of the domain	Parameters
Operating dystem	Windows 10 Pro
CPU	Intel® Core™ i7-2620 with 2.70 GHz
RAM	12 GB
Hard disk	1 TB

5 Experimental Results

Here we evaluate the experimental results of traditional machine learning algorithms i.e., KNN, LDA and SVM. Collected dataset has 3966 images. We divide the database into 80:20 split for experimenting purposes, where 80% of database is utilized for training and 20% for validation. Below Tables 3, 4 and 5 shows the category wise identification for the three classifiers KNN, LDA and SVM along with confusion matrix from Tables 6, 7 and 8 respectively.

Table 3 Category wise identification for KNN

Sl. no	Class label	Accuracy (%)
1	Insect eaten	21.5
2	Healthy	77.8
3	Leaf spot	24.8
4	Rust	51.2

Table 4 Category wise identification for LDA

Sl. no	Class label	Accuracy (%)
1	Insect eaten	29.5
2	Healthy	82.7
3	Leaf spot	43.5
4	Rust	75.3

Table 5 Category wise identification for SVM

Sl. no	Class label	Accuracy (%)
1	Insect eaten	33.8
2	Healthy	79.1
3	Leaf spot	35.9
4	Rust	73.3

Table 6 Confusion matrix for KNN

	Healthy	Insect_eaten	Leaf_spot	Rust
Healthy	319	21	48	23
Insect_eaten	27	14	19	5
Leaf spot	60	14	36	35
Rust	40	18	26	88
Average in %				57.6%

Table 7 Confusion matrix for LDA

	Healthy	Insect_eaten	Leaf_spot	Rust
Healthy	339	13	50	9
Insect_eaten	31	19	7	8
Leaf spot	42	6	63	34
Rust	19	1	23	129
Average in %				69.4%

Table 8 Confusion matrix for SVM

	Healthy	Insect_eaten	Leaf_spot	Rust
Healthy	325	26	45	15
Insect_eaten	15	22	16	12
Leaf spot	49	13	52	31
Rust	16	2	28	126
Average in %				66.2%

5.1 Experimental Results Using Deep Learning Techniques

The experimental outcomes of the basic Deep Learning model AlexNet are evaluated here. The acquired data is split 80:20, with 80 percent of the database being used for training and 20 percent being used for validation. The hyper parameters used to train the AlexNet model are listed in Table 9.

Below Tables 10 and 11 shows the category wise identification and confusion matrix obtained for AlexNet.

The graph in Fig. 7 shows the accuracy obtained for the brand new collected dataset using the AlexNet model.

Table 9 Tuning Hyper parameters

Sl. no	Name	Parameter
1	Solver type	Adam Optimizer
2	Base learning rate	0.0001
3	Batch size	64
4	Epochs	25
5	Training data	80%
6	Testing data	20%

Table 10 Category wise identification for AlexNet

Sl. no	Class label	Accuracy (%)
1	Healthy	92
2	Leaf spot	47.9
3	Insect eaten	27.8
4	Rust	63.3

Table 11 Confusion matrix for AlexNet

	Healthy	Insect_eaten	Leaf_spot	Rust
Healthy	378	9	22	2
Insect_eaten	31	18	14	2
Leaf spot	55	7	69	14
Rust	25	4	34	109
Average in %				72.8%

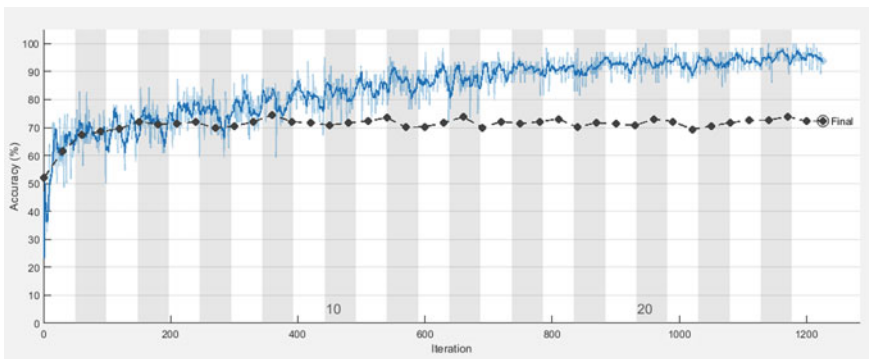


Fig. 7 Accuracy for AlexNet model using the real time dataset

6 Comparative Analysis

In this paper, we have tried to compare the traditional ML algorithms and DL techniques. In machine learning algorithms, features are taken from deep learning technique and then given as input to the ML classifier for classifying the images, as manually calculating the features and then classifying them is space and time consuming. Using this method reduces the time and space complexity which is required in ML models.

In DL model we have used pre-trained CNN architecture of AlexNet model. The models are trained on 1000’s of image categories which will help in recognizing the fungi affected leaf diseases. To get better recognizing capability we have varied the hyper parameters to achieve the mentioned accuracy of 72.8% using AlexNet model.

Below Fig. 8 shows the comparative analysis of the models and Table 12 presents the performance measures i.e., mean precision, mean recall, mean F1 score and accuracy of all the classifiers and AlexNet model.

It is observed from Table 12 that AlexNet model gives the best performance and recognition accuracy of 72.38% for the collected real time environment dataset.

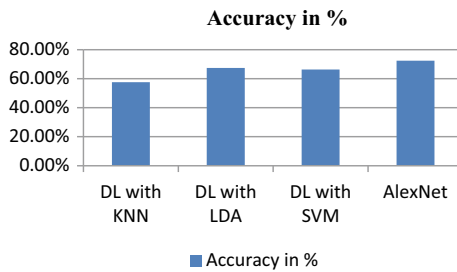


Fig. 8 Comparative analysis of models

Table 12 Performance measures: Precision, Recall, F1 score using Transfer Learning and Feature Extraction

	Precision	Recall	F1 score	Accuracy (%)
<i>Transfer Learning</i>				
AlexNet	0.6445	0.6213	0.6327	72.8
<i>Feature Extraction</i>				
KNN	0.4542	0.4632	0.4586	57.6
LDA	0.5422	0.5859	0.5632	69.4
SVM	0.5216	0.5301	0.5258	66.2

7 Conclusions

The objective of the paper is to recognize and identify 2 different types of fungi affected leaves, insect eaten leaf and healthy leaves of Guava plant. It is the necessity of the current scenario to recognize these infections at an initial stage. Using the proposed given method the time and space complexity of calculating and storing of the features is eliminated. We also eliminate the preprocessing step of removing the background of the image. The aim of this step is to determine whether eliminating this step is effective or not for classification. We have tried to work on both ML and DL techniques to get good results on the collected real time environment dataset. And from performance measures and comparative analyses it is evident that AlexNet model gave a good outcome as compared to the other classifiers.

Acknowledgements The Karnataka Science and Technology Promotion Society (KSTePS), DST, GOVT. OF KARNATAKA, has supported and funded this research.

References

1. Durmuş H, Güneş EO, Kırıcı M (2017) Disease detection on the leaves of the tomato plants by using deep learning. 2017 6th international conference on agro-geoinformatics, pp 1–5. <https://doi.org/10.1109/Agro-Geoinformatics.2017.8047016>
2. Sholihati RA, Sulistijono IA, Risnumawan A, Kusumawati E (2020) Potato leaf disease classification using deep learning approach. 2020 international electronics symposium (IES), pp 392–397. <https://doi.org/10.1109/IES50839.2020.9231784>
3. Zhang X, Qiao Y, Meng F, Fan C, Zhang M (2018) Identification of maize leaf diseases using improved deep convolutional neural networks. IEEE Access
4. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci* 7:1419. <https://doi.org/10.3389/fpls.2016.01419>
5. Arivazhagan S, Vineth Ligi S (2018) Mango leaf diseases identification using convolutional neural network. *Int J Pure Appl Math* 120(6)
6. Xiaoxiao S, Mu S, Xu Y, Cao Z, Su T (2019) Image recognition of tea leaf diseases based on convolutional neural network. arXiv preprint [arXiv:1901.02694](https://arxiv.org/abs/1901.02694)
7. Howlader MR, Habiba U, Faisal RH, Rahman MM (2019) Automatic recognition of guava leaf diseases using deep convolution neural network. 2019 international conference on electrical, computer and communication engineering (ECCE), Cox's Bazar, Bangladesh, pp 1–5
8. Gaikwad SS, Rumma S, Hangarge M (2021) Fungi classification using convolution neural network. *Turkish J Comput Mathemat Educ (TURCOMAT)* 12(10): 4563–4569
9. Sardogan M, Tuncer A, Ozen Y (2018) Plant leaf disease detection and classification based on CNN with LVQ algorithm. 3rd international conference on computer science and engineering, pp 382–385
10. Vijayakumar T (2020) Posed Inverse problem rectification using novel deep convolutional neural network. *J Innov Image Process (JIIP)*, 121–127
11. Liu B, Zhang Y, He DJ, Li Y (2017) Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* 10(1):11
12. Rothe PR, Kshirsagar RV (2015) Cotton leaf disease identification using pattern recognition techniques. In: *Pervasive Computing (ICPC)*, 2015 international conference on, pp 1–6. IEEE
13. Kai S, Xiaoyan S, Jianwei J (2007) Corn leaf disease recognition based on support vector machine method. *Transactions of the chinese society of agricultural engineering*

14. Dheir I, Abu-Naser SS (2019) Knowledge based system for diagnosing guava problems. *Int J Acad Inf Syst Res (IJASIR)* 3(3):9–15
15. Geetharamani G, Arun Pandian J (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput Electr Eng*, 323–328
16. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) Squeezenet: alexnet-level accuracy with 50x fewer parameters and <0.5MB model size. *Comput Visi Pattern Recogn*

Deep Learning Based Recognition of Plant Diseases



Swetha Parthiban, Sneha Moorthy, Sribalaji Sabanayagam, Shobana Shanmugasundaram, Athishwaran Naganathan, Mohan Annamalai, and Sabitha Balasubramanian

Abstract Every country's primary requirement is agricultural products. Infected plants have an impact on the agricultural production and economic resources of a country. Early detection of plant diseases minimizes the risk of crop loss by allowing farmers to adopt curative and preventative actions to avoid further damage. Citrus is a large plant that is primarily grown in tropical areas of the world because of its high vitamin C and other key nutrient content. Citrus plant diseases play a vital role in causing a great financial loss to the farmers and affect the economy of the country. The tomato ranks among the most economically important vegetable crops, with a global production rate of around 200 million tonnes. But due to diseases occurring in the tomato plants, the farmers are facing a great loss. To prevent these losses and provide an immediate cure, we create a model to detect plant diseases. To get a better performance model, the data augmentation process is used, and we got 18,392 images. In our model, we used a convolutional neural network (CNN) to classify the citrus and tomato leaf diseases. The diseases are Mancha Graxa, Citrus Canker, Tomato Early Blight, Tomato Septoria Leafspot, Tomato Spider Mite Two-spotted spider mite (*Tetranychus urticae*), Tomato mosaic virus (Tobamovirus), Tomato yellow leaf curl virus (Begomovirus), Tomato Target Spot (*Corynespora cassiicola*). Convolution Neural Network (CNN) is stacked by multiple convolutions and pooling layers to detect plant leaf diseases. We train a model with augmented images. After the model gets trained, it is thoroughly tested to ensure that the results are accurate. Thereby we were able to achieve high validation accuracy with a rate of (97.02%) for this dataset used. As the results showed, applying the deep CNN model to identify diseases could greatly impact the accurate identification of diseases, and could even prove useful in detecting diseases in real-time agricultural systems.

S. Parthiban (✉) · S. Moorthy · S. Sabanayagam · S. Shanmugasundaram · A. Naganathan · M. Annamalai · S. Balasubramanian
Kumaraguru College of Technology, Coimbatore, India
e-mail: swetha.20ei@kct.ac.in

S. Balasubramanian
e-mail: Sabitha.b.mce@kct.ac.in

Keywords Citrus disease · Tomato diseases · Deep learning · Convolutional Neural Network (CNN)

1 Introduction

In recent years, digital image processing, image analysis, and machine vision have evolved dramatically, and they have become a key component of artificial intelligence and the link between grounded theory and applied technology. These technologies are widely utilized in industry and medicine, but they are only seldom used in agriculture and natural ecosystems. In the domain of image processing and Deep learning, as well as its application in numerous fields of engineering, significant advances have been made. Agriculture research aims to boost food production and quality while also reducing costs and increasing profitability. Fruits are high in vitamins and minerals, as well as fibre and other non-nutrient compounds. Fruit from citrus plants contains high levels of vitamin C, which has multiple health benefits and is also used as raw material in many agricultural industries. Various diseases damage citrus fruit plants, including citrus canker and Mancha graxa. Infectious citrus canker causes yellow halo lesions as well as yellow spots on leaves and fruits. Canker bacteria spread easily and quickly through clothing and infected components to air currents, plants, birds, and humans. Greasy spot is a disease that is common in citrus cultivars produced in tropical and semitropical climates. *Mycosphaerella citri* is the cause of the sickness. On the underside of mature citrus leaves, symptoms show as yellow to dark brown to black lesions. Infected cells fail to generate chlorophyll, resulting in these yellow (chlorotic) blotches. On lemons and grapefruit, lesions are more yellowish and widespread, whereas on tangerines, lesions are more elevated and darker. Tomato Early Blight disease is caused by the fungi *Alternaria linariae*, Lesion size can be substantial, usually surrounding the entire fruit. For infected fruit, concentric rings can also be seen. Tomato Septoria leafspot is caused by the fungus *Septoria lycopersici*, older leaves reveal a large number of small, circular spots with dark borders surrounding a beige center. Tiny black flecks must be spore-producing bodies located within the centers of the spots. In the agricultural sector, the Tomato Spider mites Two-spotted spider mite is a widespread pest that affects nearly all plants (more than 1000 species). Tomato Target Spot is caused by *Corynespora cassiicola* and this disease causes severe foliar damage to greenhouse and field tomatoes. The tomato yellow leaf curl virus is part of the plant family Geminiviridae and belongs to the genus Begomovirus and throughout the tropical and subtropical regions, the disease causes severe economic losses. The tomato mosaic virus is a plant pathogenic virus and a member of the class tobamovirus. In the family tobamoviridae, it belongs to the genus Tobamovirus, and this disease is easily spread by contact, cultural practices, or contaminated seeds. It is extremely important to identify plant diseases early in the field in order to manage their detection and spread. The conventional approach to disease identification relies on agricultural extension organizations for support, but this approach is expensive and limited in countries with low resources

and infrastructure. In this case, Deep learning helps us to sought out this problem because deep learning has the advantage of requiring minimal domain expertise because no feature engineering is required. An automatic plant disease detection system based on machine learning algorithms detects diseased plants and the type of disease they have by using computer vision. Convolution neural network (CNN) can be used as a deep learning network for images to do this. CNN is used to extract visual properties such as horizontal and vertical edges, RGB values, and so on. For visual feature extraction, CNN is the finest deep learning neural network. By giving many photos of diseased plants, a CNN-based network may be trained to detect disease in plants, and the trained model can then be used to forecast disease in plants using photographs of plant leaves in the future.

2 Literature Review

Chohan et al. [1] suggested a deep learning model for detecting plant pathogens. They apply augmentation to increase the data before training the model with a convolutional neural network. They obtain an accuracy of over 98.3%. Senthilkumar et al. [2] presented a method for identifying and classifying citrus plant diseases it constitutes of 4 processes that is pre-processing, segmentation, feature extraction, and classification. Otsu is used for segmentation. Inception ResNet is used for feature extraction. For classification, they used random forests. They got 98.91% accuracy. Zeng et al. [3] developed a deep learning model to detect citrus plant diseases with help of inception_v3 architecture in addition to that they used GANs based data augmentation to increase the performance. Huang et al. [4] proposed two models one to detect plant disease and other is to plant classification. They obtain accuracy around 98% for plant classification and 87% for disease identification. Kukreja et al. [5] proposed a deep learning model for the detection of citrus fruit diseases using the process of data augmentation and pre-processing. They attain accuracy around 89.1%. Mohanty et al. [6] trained a deep convolutional neural network to 12 to identify 14 crop species and 26 diseases with 54,306 images and paved the way for smartphone-assisted disease diagnosis. They obtain accuracy around 99.35%. Song et al. [7] proposed a model to detect citrus plant diseases using YOLO (you only look once) algorithm. YOLO can detect and circle around a disease in real time on an image or video. Jasim et al. [8] proposed a robust methodology to detect and classify the diseases. They obtain accuracy around 98%. Arya et al. [9] have analyzed the various research works and reviewed the architecture used for the detection of diseases in plant leaves. Alruwaili et al. [10] used the AlexNet architecture and it was customized in order to detect disease in an efficient manner. Then they use new augmentation method. To overcome the overfitting, they used “sgdm” method.

3 Scope

The scope of our project is to maximize productivity and ensure agricultural sustainability. This model will serve as a benefit for the farmers who have cultivated citrus and tomato plants. Farmers get benefitted from this model with high yield and minimize the economic loss and it gives a positive impact on the farmers.

4 Methodology

4.1 Procedure

1. Datasets

In computing, a dataset is a collection of related, discrete pieces of related information that can be gotten at exclusively or mix-and-matched, or that can be tracked in aggregate. Figure 1 shows the diseased leaf images.

2. Data Augmentation

Adding augmentation data to existing data sets or creating new synthetic data sets from existing data is a process of increasing the amount of data in data analysis. This function serves as a regularizer and reduces overfitting when training a model. It has a lot to do with data analysis oversampling. Through the process of data augmentation, we have increased our original dataset to double the size. Consequently, we were able to create adequate datasets and train our model to achieve maximum accuracy.



Fig. 1 Leaf disease image

Table 1 I: CNN training parameters

Parameter	Value
Epochs	25
Batch size	32
Activation in middle layer	ReLU
Activation in final layer	SoftMax

3. Environment

Execution of code needs an access to a system with GPU. For this, we used an online platform called Google colab as these notebooks have GPU access. To detect the citrus and tomato diseases, we use Convolutional Neural Network Algorithm.

4. Convolutional Neural Network Algorithm

The Input layer, Middle layers, and Output layer make up a Convolutional Neural Network, or CNN. The input layer is responsible for accepting features such as input, or images. The number of middle layer nodes is determined by the application. A result is created by the output layer

Convolutional Layer-Convolution is performed on the pixel values in conjunction with the kernel matrix. The kernel matrix is dragged across the pixel matrix to calculate the result.

Max Pooling Layer-Filter maps are generated using this layer to minimize their size. This reduces the likelihood of overfitting issues.

Rectified Linear Units (ReLU) Activation Function-With the Rectified Linear Units (ReLU) Activation Function, all negative values are replaced by zero (0) while all positive values are maintained.

Fully Connected Layer-A node from the previous hidden layer is connected to a node from the next hidden layer in this layer, which is also known as Dense Nodes. Each layer will be connected by use of edge connections among the neurons. Here Table 1 shows the training parameters of the CNN model.

4.2 Library

Keras is a popular deep learning library that runs on top of other libraries such as tensor flow. Its minimal Python structure facilitates learning and quick writing of deep learning models.

Algorithm

STEP 1: Importing the libraries and initializing the data path.

STEP 2: Importing the datasets.

STEP 3: preprocessing—write a function to resize the dataset images to (224,224) so that they can be fit for training. Also visualizing our data and dividing it into training and testing categories.

STEP 4: Defining train and test data into the model. The train dataset has 12,723 images belonging to 8 classes and the test dataset has 5669 images belonging to 8 classes.

STEP 5: we use keras library to develop a plant disease detection model. Because it has good support for functions that enable users to deploy models fast. Keras supports building neural network models with fewer lines of code.

STEP 6: we create a sequential model for this plant disease classification. we develop the 1st block of 2D convolutional layers with 32 filters of 5*5 kernel and using ReLU (rectified linear unit) activation function then we perform max pooling operation in the layers. Secondly, we used 32 filters of the 3*3 kernel with ReLU (rectified linear units) activation function in the layers and then we performed maximum pooling. In the 3rd block of 2D convolutional layers, we have 64 filters of 3*3 kernels, one of which is active in a ReLU (rectified linear unit) activation function, and the other operates in a Max Pooling mode.

The next layer is a flattened layer. We used 25% (0.25) dropout function in the dense layer.

STEP 7: Here, we declare all the hyper parameters required for our plant disease classification such as epochs, steps, batch size, learning rate etc. Fit our training data to our model and set the batch size to 32, which will accept 398 values at a time until all parameters are met. The number of epochs here refers to the number of times it will be processed.

STEP 8: Training the Model-The model has been developed, and it should be compiled using the Adam optimizer, which is one of the most familiar optimizers.

STEP 9: We take 25 epochs to train our model. More epochs increase the accuracy and decrease the loss. we can see here that accuracy in each step increase. we train our model 25 times. When test images train with lots of train images then accuracy will increase, and our model predicts more accuracy. More accuracy means our model works properly and gives desired output. Figure 2 gives the summary of CNN model architecture.

STEP 10: Detection of the disease when the input is given.

STEP 11: Computing Loss Calculation Based on Training and Test Results.

STEP 12: Accuracy Calculation for Training and Test Results-Matplotlib was used to show the accuracy plot on the training and test sets. A simple graph analysis can clearly reveal the difference in training and test set accuracy.

Accuracy = Number of accurately predicted records/Total number of records.

```

[ ] Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d_1 (Conv2D)           (None, 220, 220, 32)       2432
-----
max_pooling2d_1 (MaxPooling2 (None, 73, 73, 32)         0
-----
conv2d_2 (Conv2D)           (None, 71, 71, 32)         9248
-----
max_pooling2d_2 (MaxPooling2 (None, 35, 35, 32)         0
-----
conv2d_3 (Conv2D)           (None, 33, 33, 64)         18496
-----
max_pooling2d_3 (MaxPooling2 (None, 16, 16, 64)         0
-----
flatten_1 (Flatten)         (None, 16384)              0
-----
dense (Dense)                (None, 512)                8389120
-----
dropout (Dropout)           (None, 512)                0
-----
dense_1 (Dense)              (None, 128)                65664
-----
dense_2 (Dense)              (None, 8)                  1032
-----
Total params: 8,485,992
Trainable params: 8,485,992
Non-trainable params: 0
-----
    
```

Fig. 2 Convolutional layers

The result is 0.9702, which means the model is 97.02% accurate in making a correct prediction. The flow of the project is pictured in Fig. 3.

5 Results and Findings

This study emphasizes the need of early diagnosis of plant diseases. Deep Learning was used to create this model. The accuracy of this model was tested using 30% (5669) of photos. The photographs in this gallery are from eight distinct classes. For testing, 30% of each class was chosen at random. The correctness of the testing dataset is greater than 95%. Model properly Classified 5669 photos out of 18,392 images. On the testing dataset, our model generated the Training and Validation accuracy curve as shown in Figs. 4 and 5.

FLOWCHART.

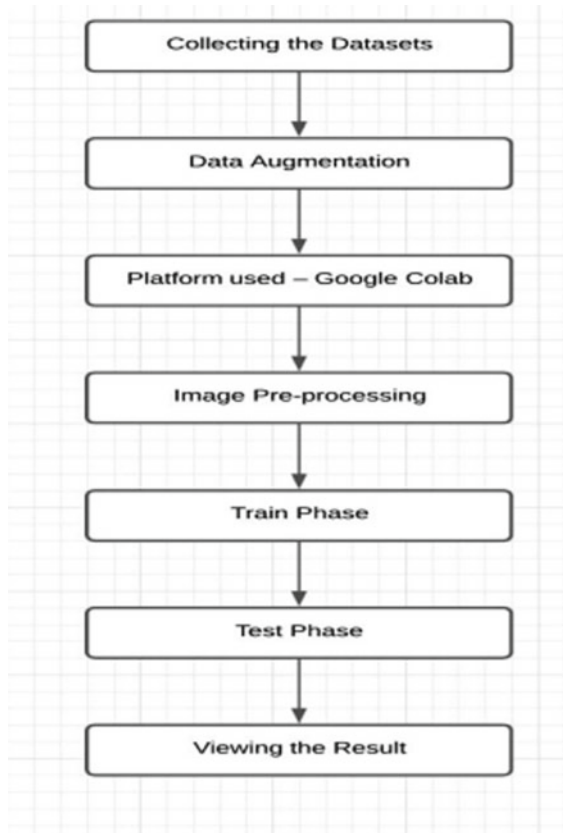


Fig. 3 Work flow

6 Discussion

To identify citrus and tomato illness with high efficiency, this method primarily uses Convolutional Neural Networks. When verification performance and model storage are constrained, the CNN model can be useful. An important component for diagnosing and identifying crop diseases and insect pests is the availability of rich and diverse databases, and a large dataset can decrease the error rate in the recognition process. The power of deep learning tools can only be fully unleashed by accumulating. In addition, we intend to acquire ever larger databases in order to improve the model’s generalization capabilities. As the dataset collected on citrus diseases is being analyzed, the backgrounds can be removed, allowing better disease images to be used as training models for improving the accuracy of disease identification. It is possible

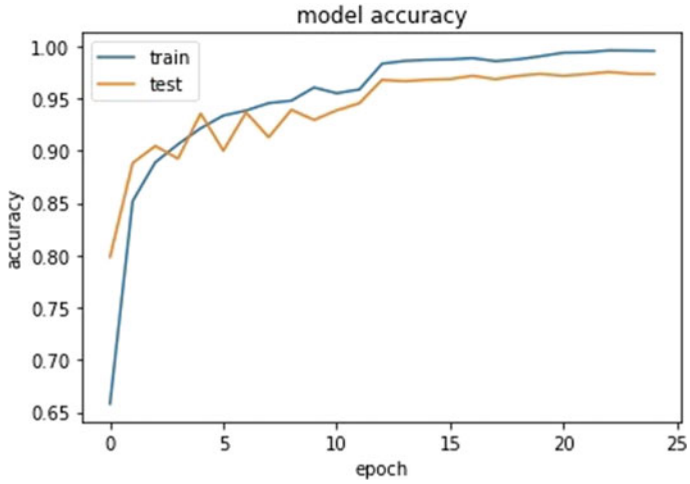


Fig. 4 Model accuracy

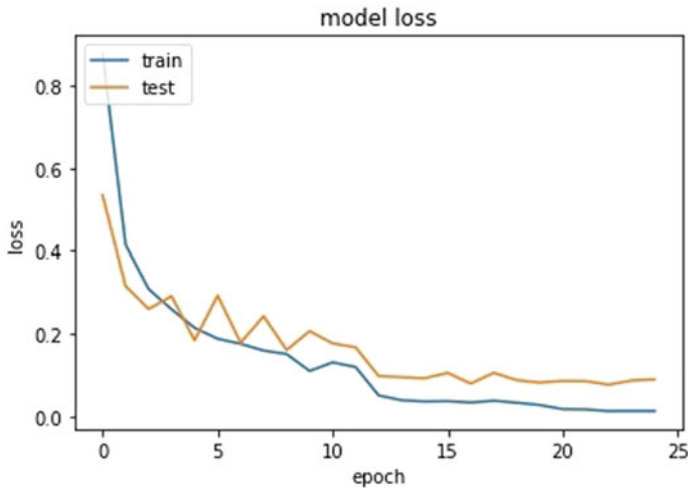


Fig. 5 Model loss

to attain identification accuracy by optimizing the convolution network algorithm. Optimisation of the convolution operation can ensure more accurate identification of data obtained from larger datasets.

7 Conclusion

Using a form of modern approaches, plant diseases are detected and classified using sick leaves. Commercial solutions to spot these diseases aren't available at the current time. In our work, we used CNN models for the detection of plant diseases using diseased-leaf images of citrus and *Lycopersicon esculentum*. The quality dataset with 18,392 images was accustomed to train and test the model (after data augmentation). There are 8 different classes of plant diseases namely Mancha Graxa, Citrus Canker, Tomato Early Blight, Tomato Septoria Leafspot, Tomato mite Two-spotted mite (*Tetranychus urticae*), Tomato mosaic virus (Tobamovirus), Tomato yellow leaf curl virus (Begomovirus), Tomato Target Spot (*Corynespora cassiicola*).

After splitting the dataset into 70–30 (70% data for training, 30% data for testing), 97.02% accuracy was achieved in our model. On average, it took 25 and 25 s/epoch, respectively, on coloured images. With relevance to both accuracy and loss, the implemented deep-learning model reduces the deviation from the best deep learning model. The desired time to coach the model was much but that of other machine-learning approaches.

8 Future Scope

The present model is modified to measure the detection model and it is designed as an internet application which helps farmers to make a decision on the particular quantity for pesticide applications to scale back the price and environmental pollution and increase the productivity. Moreover, the system predicts the kind of plant and kind of disease so scientists can recommend specific pesticides or fungicides to be used, thus providing useful suggestions for future research. Detecting a disease during this crop is predicated solely on the leaf of the plant. Roots, stems, and branches from the crop are easier to handle if they will be included yet which increases the detection accuracy. It'll also show the disease name if the model receives input aside from leaf images. And also had a concept to incorporate another crop diseases to extend the predictability.

9 Recommendation

There are countless new plant diseases reported annually or an old disease that has reappeared and is now affecting crops previously unaffected. But the notice towards the detection of plant diseases in an early stage isn't for sure. The fundamental thought when the plant gets affected is to use fertilizers and a few chemicals utilizers but while taking concern about the health of the plant one of all the most effective ways is early detection. In the event of controversy during the growing process, ask

questions or keep an eye fixed on the matter. By following the recommendations, the impact of plant diseases is greatly reduced, leading to a decrease in shrinkage and increased profit.

References

1. Chohan M, Khan A, Chohan R, Hassan S, Mahar M (2020) Plant disease detection using deep learning. *Int J Recent Technol Eng* 9(1):909–914
2. Senthilkumar C, Kamarasan M, A novel citrus disease detection and classification using deep learning based inception resnet V2 model
3. Zeng Q, Ma X, Cheng B, Zhou E, Pang W (2020) GANs-data augmentation for citrus disease severity detection using deep learning. *IEEE Access* 8:172882–172891
4. Huang S, Liu W, Qi F, Yang K (2019) Development and validation of a deep learning algorithm for the Recognition of plant disease. In: 2019 IEEE 21st international conference on high performance computing and communications; IEEE 17th international conference on smart city; IEEE 5th international conference on data science and systems (HPCC/Smart City/DSS). IEEE, pp 1951–1957
5. Kukreja V, Dhiman P (2020) A deep neural network based disease detection scheme for citrus fruits. In: 2020 international conference on smart electronics and communication (ICOSEC). IEEE, pp 97–101
6. Mohanty SP, Hughes DP, Marcel S (2016) Using deep learning for image-based plant disease detection. *Front Plant Sci* 7:1419
7. Song C, Wang C, Yang Y (2020) Automation detection and image recognition of precision agriculture for citrus diseases. In: 2020 IEEE Eurasia conference on IOT, communication and engineering (ECICE). IEEE, vol 23, pp 187–190
8. Jasim MA, AL-Tuwaijari JM (2020) Plant leaf diseases detection and classification using image processing and deep learning techniques. In: 2020 international conference on computer science and software engineering (CSASE). IEEE, pp 259–265
9. Arya S, Singh R (2019) An analysis of deep learning techniques for plant leaf disease detection. *Int J Comput Sci Inf Secur (IJCSIS)* 17(7)
10. Alruwaili M, Abd El-Ghany S, Shehab A (2019) An enhanced plant disease classifier model based on deep learning techniques. *Int J Eng Adv Technol* 9(1):7159–7164

Artificial Cognition of Temporal Events Using Recurrent Point Process Networks



N. Bala Sundara Ganapathy and M. Deeptavarna

Abstract Data generated by real-world events are usually obtained by condensing events into continuous signals at specific intervals, which are handled by mechanisms that make this data, a part of a continuous time function. Past models have used a parametric function that relates chronological data with event sequences, which are elicited through conditional intensity functions in order to quantify the frequency of event occurrence. Thus, the resulting data is devoid of a relation between present and past event occurrences, which is more often than not, a specific task that has been determined based on prior knowledge. This particular limitation is overcome using point process networks in Recurrent Neural Network (RNN) to which this paper on the said subject will enumerate a theoretical distinction along with practically implemented instances of RNNs where one RNN is concerned with capturing the relationship among the event sequences obtained as a result of the processes described earlier, while another is employed to update the intensity function based on time series. Finally, in order to prove the relation among data in a regular interval and in a random event an attempt to discover any anomaly or deviation from prediction is also implemented.

Keywords Parametric function · Recurrent neural network · Anomaly detection · Data and model integration · Self-parameterized model creation

1 Introduction

Sequences usually result in two kinds of indicators chronologically, synchronous and asynchronous [1]. Asynchronous events are prevalent in various fields such as social networks, hospital management systems and other daily monitored tasks. Event data that is obtained does not only contain the event characteristics but also a timestamp that is denoted as:

N. Bala Sundara Ganapathy (✉) · M. Deeptavarna
Panimalar Engineering College, Chennai, India
e-mail: balabsg@gmail.com

$$\{Z_i, T_i\}N_i = 1 \quad (1)$$

which denotes the occurrence of an event. Studies focusing on the dynamics of systems which explore the timestamp relation in order to conclusively create a relationship among the data that exist, show that timestamps denote event sequences that occur in asynchronous events are fundamentally different from those of synchronous events, this is because asynchronous timestamps provide a basic insight into the network dynamic while the time aspect is deterministic. Though, the time series data provides continuous updating of background events that occur in temporal point process, many models employ time series data along with point process data. Past models treat these two processes independently [2–5] ignoring the influences they hold over each other, there is a need to view them as a closely related process, these are called joint models, where the dynamics of point process and the series of timestamps are taken together. These “Joint models” though are rare and almost inexistent.

The proposed work through this project is one such effort to create a “Joint model” that interlinks the time series and event sequence to each other. The most popular method of doing so is by detecting multiple events from series data. Also, Statistical aggregation or frequency counts are often performed on each time interval of equal length, while this can be considered a crude method as it can cause loss of information about the actual functioning of the process at a very inceptive stage. This has been offset by studying the recent works on modeling point process which include mathematical reformulations and optimization techniques along with novel parametric forms that strive to play the part of prior human knowledge that has already been shown to have been the governing aspect in forming the parametric rules earlier [6–8]. A major limitation that these models exhibit is in their capability to capture the dynamics of data and misspecification.

This led to the development of a model that functions in the absence of parametric regulations. But these models are under Hawkes’s process formulation, which bases its results on risks of unknown model complexity and is not congenial for point processes that do not follow the self-excitation or the mutual excitation of nodes which is inherent in the Hawkes model [7–9]. In another recent work, the authors described a semi-parametric pseudo-point process model, which employs a time-decaying influence between events and the background of intensity is constant and the event type is regarded as a mark associated with univariate process. In papers proposed on this subject, contingent force of a point interaction as nonlinear planning from the installing of time series and previous occasion information to the anticipated transient event power of future occasions with various kinds, a revised version of “Joint model”. This was done to accommodate and create a model capable of handling all paradigms of real event data both synchronously and asynchronously developed data. In order to eliminate a parametric form of intensity function the event time is directly applied into the model. RNN are chosen as the model for such non-parametric mapping, by utilizing deep learning techniques to increase the efficacy of the proposed model in creating an intensity function of the point process. Thus, the result of such pursuits and combination of studies in such approaches

proposes a synergistic multi-RNN, as the choice for non linear and dynamic mapping without prior mapping. As mentioned, in order to appropriately monitor the developments, a mechanism to check anomalies are also infused to improve its capability for prediction and relational mining.

2 Recurrent Point Process Network

2.1 Recurrent Neural Network

The main component of the proposed model is the Recurrent Neural Networks (RNNs) and its advanced variations like Long Short Term Memory units and Gated Recurrent Units. RNNs are dynamical frameworks whose next state and yield rely upon the current network state and its input information, which are more broad models than the feed-forward networks. RNNs have for some time been investigated in perceptual applications for a long time; anyway, it tends to be undeniably challenging for preparing RNNs to learn long-range elements to some degree because of the disappearing and detonating inclinations issue. LSTMs give an answer by joining memory units that permit the network to realize when to fail to remember past secret states and when to refresh covered up states given new data. In various language problems, vision and speech, the RNN's and LSTM's models have been successfully applied recently. There are two main aspects that have been considered in this paper in the application point of view. They are,

- i. Synchronized Events—Sequence of events with equally spaced intervals.
- ii. Asynchronous Events—Sequence of events with timestamps about their occurrence, which are randomly distributed over the continuous time series.

2.2 Interoperability and Prediction Module

Prediction precision and model interpretability are two objectives of numerous fruitful prescient strategies. Existing works regularly need to experience the tradeoff between the two by either picking complex discovery models like deep neural networks or depending on conventional models with a better translation, for example, Logistic Regression with less exactness contrasted and best in class deep neural network models. Notwithstanding the promising increase in exactness, RNNs are somewhat hard to decipher. There have been a few endeavors to decipher RNNs, and provide a generic method of interpreting a model but owing to the flexibility that is inherent in RNN it usually doesn't seem feasible when a generic interpretation is expanded to understand all other models. The system proposed makes use of another model that works closely to interpret the event triggers by monitoring the data variation from the predicted values which is called the attention model whereas a simple

mathematical equation that observes the loss and cost at each stage enables us to ensure interoperability without any kind of trade-off between them and maximizing their combined efficacy.

2.3 Point Processes Equations

Point process is a mathematically effective framework for modeling synchronous and asynchronous event data. It is an arbitrary process whose realization comprises of a rundown of discrete occasions restricted on schedule. The elements of the point process can be very much caught by its contingent intensity function, whose definition is momentarily investigated here,

$$[t, t + dt], \lambda(t) \tag{2}$$

The Eq. (2) represents the rate of occurrence of a new event with respective conditions on the history denoted as follows:

$$Ht = \{z_i, t_i | t_i < t\} \tag{3}$$

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} E(N(t + \Delta t) - N(t) | Ht) \tag{4}$$

$$\Delta t = E(dN(t) | Ht) dt. \tag{5}$$

Equation (5) calculates the number of expected events that happened in the interval $[t, t + dt]$, given the historical observations Ht . The conditional intensity function played a vital role in the point processes.

3 Automated Data Relation Process

3.1 Model Creation

When a dataset containing regular intervals of events are sent through the model, the data is first broken down into batches and a few parameters such as data size, encapsulations and data format are considered. This paves the way for the model, as this information largely dictates the most appropriate model that can be employed in order to attain the required weights and bias in the equation of the model constructed.

Since the model is highly susceptible to vanishing gradient and cascading gradient effects, LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) are incorporated. The model is capable of deciding on the best alternative in order to

avoid gradient problems. LSTM and GRU are of special importance since data sets that are given as input are not subjected to manual inference methods which creates a need for the model to be capable of creating its own model based on pre-programmed formulas that denote the number of layers and hidden nodes. While each unit can simply be a recurrent iteration of the first node in the second layer, the input and output layers are designed automatically based on input data sequences, which are a result of internal data processing which sequences the large input data into batches and sequences for more optimal routes. Finally, the entire model is subjected to an optimizer which ensures no regressions are found.

3.2 Training Phase

Once the data is sent to the training phase of the model, each epoch, their loss combined with total loss, beta, f-values, and weights are calculated and stored as checkpoints. These checkpoints can be considered as the model's inference of the data given. During the training phase, the losses are calculated and a cumulative loss value is passed as the total loss for the entire epoch of data that is learned by the model. We first train the model with a dataset that has no anomaly and then use the model to identify anomalies in a test set where anomalies are present. In order to prove that a relation among the data is also created, this inference is passed to the attention module described earlier in this paper.

Since multi-step predictions using RNN is a complex problem due to the accumulation of prediction errors as the process progresses causing inaccurate results. To prevent such issues the model is incorporated with noise handling capacities such as Teacher Forcing Loss and Free Running Loss, which are referred to as loss in the execution phase of the model.

3.3 Testing Phase

The model tests a part of the entire dataset in order to fine tune its own weights and bias, also called as cost or loss depending upon how the deviation is calculated. This allows the model to learn and correct itself which is of paramount importance when trying to predict the future event occurrences which follows this phase.

3.4 Anomaly Detection

The checkpoints allow the model to construct predictions based on past events, and these predictions are placed at 1-step and 10-step intervals. The data deviating from these predictions at a particular step interval is considered an anomaly which

is interpreted as an anomaly in the event plane. This data is passed through the network, where values of mean, beta, predictions, f-value and step-predictions are carried out and saved as checkpoints so that the data need not be trained every time for creating instances and can be simply interpreted from saved checkpoints. These step predictions are carried out using the following mechanism.

3.5 Visualization

To allow a visual representation of said anomaly, predictions and the mean, the entire data along with parameters such as mean, predictions, and deviations are mapped onto a graph. This allows for a human inference from the entire data, which is in part the goal of the model.

4 Temporal Variation Samples

4.1 Synchronous Event

When dealing with variable inputs such as the models previously used [9]. There is a clear need for distinctions from those used in asynchronous events [9, 10]. To elicit the working of the model when faced with a temporal event that shows a recurring pattern, the model is expected to recognize this and prove this recognition through the detection of any anomaly if at all present in the data stream. For this purpose, we employed the dataset of an ECG chart and processed them into a stream of acceptable data format with the help of previous studies [11–13]. This is a result of pickling the input image and then serializing it to generate a data stream of numerical value from the image data. Input to the model is directly given as the image which is later subjected to serialization that converts the image into a sequence of values. The image in its actuality is devoid of any errors or anomalies which are used to train the model, while for testing the same image can be given as input with a few anomalies introduced or another entirely new image containing anomalies can be used. The model then proceeds to process the values which result after serialization of the image. These results in the creation of overlapped prediction sets that are comparable with the mean in order to show the deviation or any forms of anomaly that might exist. Figure 1 is the enlarged portion of the abnormal variation region.

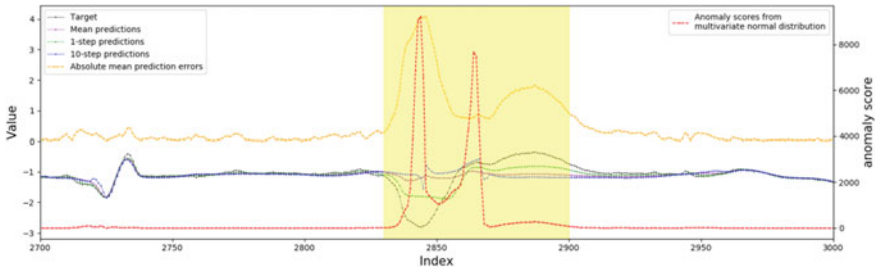


Fig. 1 Anomaly detection on ECG dataset

4.2 Asynchronous Event

While distinguishing a temporal pattern is simply a subject of waiting for the pattern to repeat in a synchronous temporal event, asynchronous events differ quite extensively in this subject of behavior. In order to deal with such scenarios, we take upon values of covariance and mean into a Poisson or Hawkes model. The model is bolstered by past instances and examples of successful implementation of Hawkes model in scenarios of temporal randomness or irregularity in occurrences that are usually referred to as events [14]. After applying serialization in a manner similar to the one carried out in the ECG charts for obtaining a series of numerical inputs, the data on NYC taxi dispatches and gesture data were converted into inputs. Since the model does not discriminate nor does it have the capacity to distinguish between a temporal event of repetitive pattern and irregular pattern, the model tries to elicit regions of deviation from predicted values thereby creating an acknowledgement of event relation creation. Figure 2 shows the deviation from predicted values.

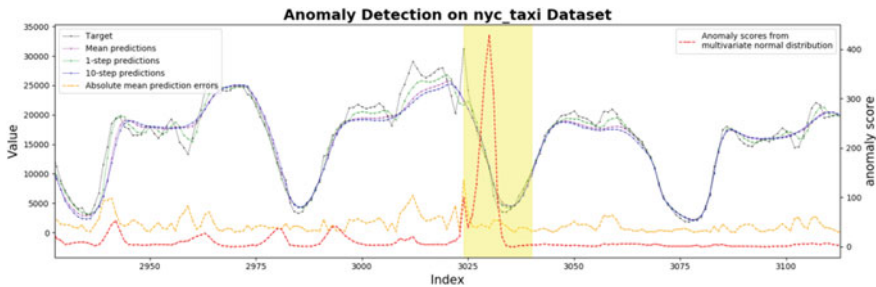


Fig. 2 Anomaly detection on NYC Taxi dataset

5 Conclusion

Thus, a model that is capable of autonomously handling varying data sizes and seamlessly integrates data and model integration without any prior parametric limitations is developed. The model also includes a self-parameterized model creation making manual parameter specification redundant. The event relation is elicited using an attention module that detects anomalies proving the model's capabilities and also the model's advanced efficacy is proved using various data samples comprising of both synthetic and real-life data sets.

References

1. Deshpande P, Marathe K, De A, Sarawagi S (2021) Long horizon forecasting with temporal point processes. *ACM international conference on web search and data mining*, pp 571–579
2. Lugo L, Moreno JG, Hubert G (2021) Modeling user search tasks with a language-agnostic unsupervised approach. *Advances in Information Retrieval*, Springer, pp 405–418
3. Ji G, Zhu Y, Niu Y, Hu K (2021) Classification and evaluation for microblog popularity prediction. *J Phys*. IOP Publishing Ltd, pp 1–10
4. Baggio JA (2021) Knowledge generation via social-knowledge network co-evolution: 30 years (1990–2019) of adaptation, mitigation and transformation related to climate change. *Clim Change* 167:13. <https://doi.org/10.1007/s10584-021-03146-5>
5. Cai H, Nguyen TT, Li Y et al (2020) Modeling marked temporal point process using multi-relation structure RNN. *CognComput* 12:499–512
6. Bernis G, Scotti S (2020) Clustering effects via Hawkes processes. In: Jiao Y (eds) *From probability to finance*. *Mathematical Lectures from Peking University*. Springer, Singapore. https://doi.org/10.1007/978-981-15-1576-7_3
7. Shi L, Lu P, Yan J (2020) Causality learning from time series data for the industrial finance analysis via the multi-dimensional point process. *IASC* 26(5):873–885
8. Zhou K, Zha H, Song L (2013) Learning triggering kernels for multi-dimensional Hawkes processes. *ICML* 3:1301–1309
9. El-Bouri R, Taylor T, Youssef A, Zhu T, Clifton DA (2021) Machine learning in patient flow: a review. *Progress Biomed Eng* 3(2)
10. Graves A, Rahman Mohamed A, Hinton G (2014) Towards end-to-end speech recognition with recurrent neural networks. In *ICML, JMLR:W&CP vol 32*, pp 1764–1772
11. Zhang L-N, Liu J-W, Zuo X (2020) Survival analysis of failures based on Hawkes process with Weibull base intensity. *Eng Appl Artif Intell* 93
12. Graves A (2013) Generating sequences with recurrent neural networks. [arXiv:1308.0850](https://arxiv.org/abs/1308.0850)
13. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. [arXiv:1412.3555](https://arxiv.org/abs/1412.3555)
14. Kim M (2020) Understanding time-evolving citation dynamics across fields of sciences. *Appl Sci* 10:5846. <https://doi.org/10.3390/app10175846>

Performance Analysis of Energy Efficient Video Transmission Using LEACH Based Protocol in WSN



M. Radhika and P. Sivakumar

Abstract Video transmission over wireless sensor networks is an on-demand research due to the advantage of low power consumption and low implementation cost. Video transmission has deployed in many of the WMSN trends, which includes multimedia, video surveillance, health-care monitoring, military, environmental and industrial applications. In this paper, an energy efficient video transmission using hierarchical LEACH based routing protocol has been conducted through MATLAB simulation, based on H.264 video coding technique. In this investigation of video transmission over LEACH based routing protocols, our proposed modified micro genetic algorithm consumes less energy when compared with other LEACH variants.

Keywords Energy efficient routing protocol · Wireless sensor network · Video transmission · Genetic algorithm · LEACH

1 Introduction

Wireless sensor networks (WSN) are the most leading, talented and emerging equipment that plays a significant role in many applications like environmental monitoring, traffic controlling and monitoring [1], video surveillance, agriculture, agriculture [2], multimedia surveillance [3], military and industrial applications. Wireless sensor network is a self-organized network which consists of four fundamental parameters: sensors, clusters, interconnecting network, and computing resources.

WSN has many challenges [4–16]: energy efficiency, network lifetime, QoS, scalability, maintainability, robustness, and security. For the past few decades, researchers developed a lot of innovative protocols to overcome the challenges in WSN, especially the lifetime of the network. In our previous work [17, 18], micro genetic based

M. Radhika (✉)

Department of Information and Communication Engineering, Anna University, Chennai, Tamilnadu, India

P. Sivakumar

Department of Electronics and Communication Engineering, Dr. N.G.P. Institute of Technology, Coimbatore, Tamilnadu, India

LEACH protocol was proposed and the result shows an improvement in network lifetime by 20% to 94% when compared with other LEACH variants.

Multimedia transmission over WSN [19, 20] is the most challenging task due to the availability of minimum power sources. In this paper, we proposed a video transmission analysis over LEACH protocols.

2 Related Work

Zigbee-based video streaming in WSN [21] achieved an acceptable video quality of the network. Video packets are transmitted to analyze the quality of service (QoS) [22, 23]. Cross-layer communications with multipath routing techniques were proposed by [24], and produced a good result with acceptable QoS and network lifetime improvement. Multi-path content-aware video transmission [25] has a control signal which will periodically update and sends a message from source to sink node to exchange the routing path information. Based on the priority and energy level, the packets are selected for video compression in the EQV architecture [26]. It is one of the energy efficient high-quality video transmissions over WSN in the cross layer architecture.

3 Problem Statement and Our Contribution

WSN contains non-rechargeable battery power at each and every node. Hence, power consumption is an important factor to improve the lifetime of the network. In general, transmitting a video over LEACH based hierarchical protocol consumes more power. Here, we introduced an energy efficient routing protocol to improve the lifetime of network by consuming less energy. Our proposed micro genetic algorithm (μ GA) based LEACH protocol consumes lesser energy of 35% to 84% when compared with other LEACH variants like LEACH, LEACH-C, LEACH-Genetic Algorithm (GA), and GADA-LEACH protocol.

4 LEACH Based Routing Protocol

LEACH is the first hierarchical protocol proposed by [27], in which the cluster heads and cluster members are selected based on the set-up phase and steady-state phase. This choice of cluster heads (CHs) may not aware of the node location. Later, LEACH-C [28] was introduced which is similar to LEACH contains set-up and steady-state phase, but node location is taken into account for the selection of cluster head is an advantage over LEACH. In the LEACH-GA [29], a genetic algorithm is employed for the selection of optimal cluster head probability. The

preparation phase is introduced at the beginning of the set-up and steady-state phase. GADA-LEACH [30], relay nodes are introduced in between the base station and cluster head for long distance communication. GADA-LEACH improves the network lifetime when compared to LEACH, LEACH-C, and LEACH-GA protocols. In our previous work [17], the μ GA-LEACH protocol was implemented and our result has concluded that μ GA-LEACH improves the lifetime of network by 94%, 91%, 54%, and 20% greater than the LEACH, LEACH-C, LEACH-Genetic Algorithm (GA), and GADA-LEACH protocol respectively. The overall process of the proposed μ GA based LEACH protocol is shown in below Fig. 1. The network lifetime comparisons of the proposed protocol with the other LEACH variant are shown in Fig. 2.

5 Video Transmission Over WSN

Video transmission is the most challenging task in wireless sensor networks, which includes the following steps.

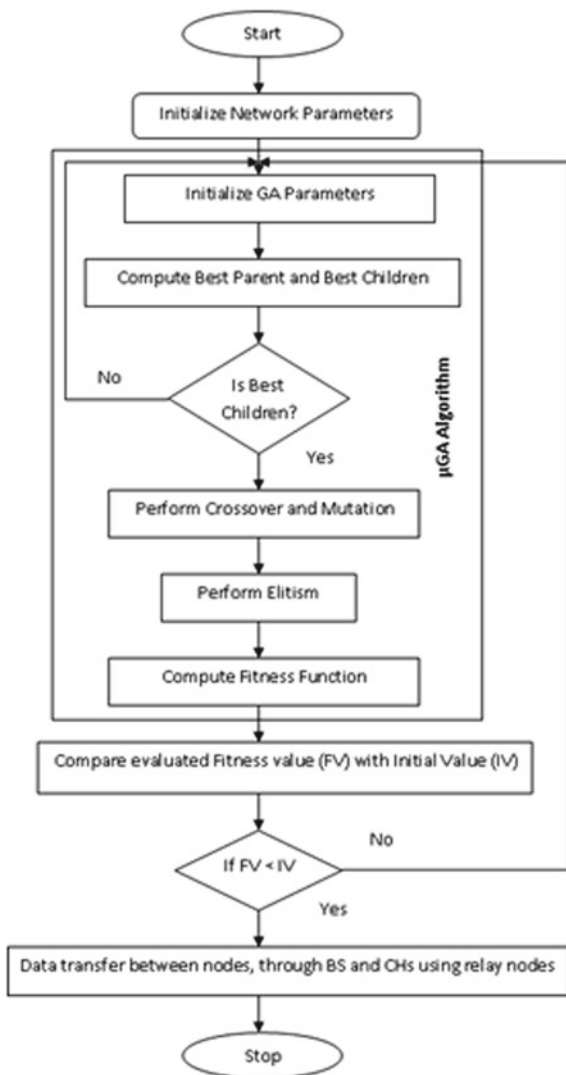
- Generate 'N' number of nodes. Among that total, select 2 nodes for making a transmission.
- Initialize the genetic algorithm and network parameters
- Select the protocol to perform the video transmission over the network.
- The video packets are converted into a sequence for binary bitstreams, and then the binary sequences are transmitted through the selected protocol in the constructed network using H.264/AVC encoding technique.
- After transmitting, the packets are reconstructed using H.264/AVC decoding technique.
- At last, the total energy consumed by the network is calculated.

6 Experimental Result

In this section, simulation of video transmission in wireless sensor network was analyzed on MATLAB tool with the network area of $100 \times 100 \text{ m}^2$. Let us consider the total number of nodes are 100, which are randomly distributed through the network and the base station is situated at the center of the network (50, 50). Among that total node, two nodes are picked up randomly for video transmission. In which one will act as a source and the other is the destination node. In Fig. 3, dark pink dashed line represents the path for video transmission between the source and destination.

For the simulation, ant-maze video [31] has been considered with different frame lengths and initial energies as shown in Fig. 4. The simulation parameters are shown in Table 1. First radio model is used for the network which was proposed by Heinzelman et al. [31]. The energy consumption for the cluster heads is given in Eq. 1,

Fig. 1 Proposed μ GA-LEACH protocol [17]



$$E_{CH}(l,d) = \begin{cases} l \times [E_{elec}(\frac{n}{k} - 1) + E_{DA}\frac{n}{k} + E_{elec} + \varepsilon_{fs} \times d_{toBS}^2] \text{ if } d_{toBS} < d_0 \\ l \times [E_{elec}(\frac{n}{k} - 1) + E_{DA}\frac{n}{k} + E_{elec} + \varepsilon_{mp} \times d_{toBS}^4] \text{ if } d_{toBS} \geq d_0 \end{cases} \quad (1)$$

Energy consumption by the non-CHs is given in Eq. 2,

$$E_{non-CH}(l,d) = l \times E_{elec} + l \times \varepsilon_{fs} \times d_{toCH}^2 \quad (2)$$

Fig. 2 With the initial energy of 0.2 J/node [17]

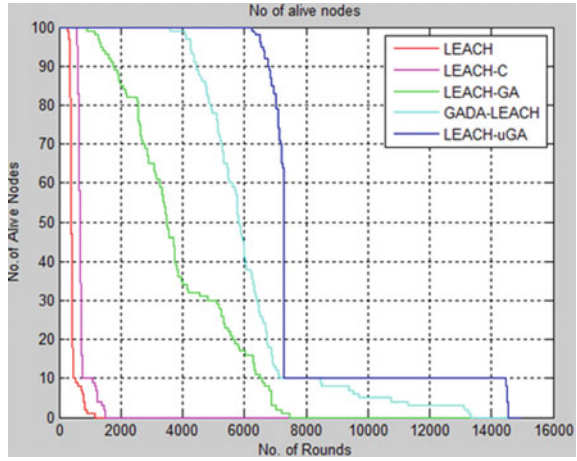
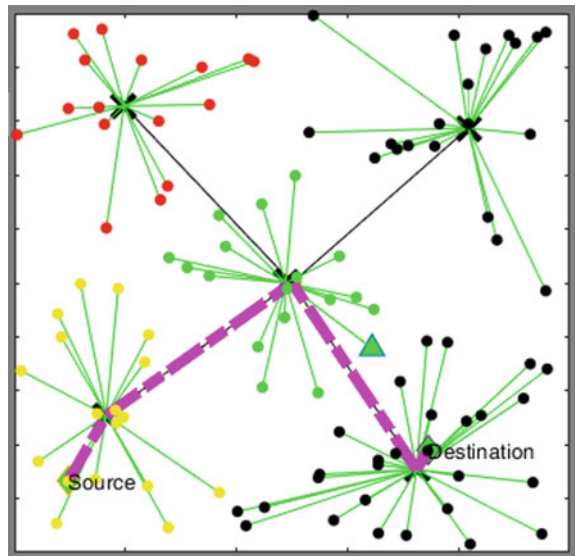


Fig. 3 Network Structure



$$d_0 = \sqrt{\frac{\epsilon f s}{s_{mp}}} \tag{3}$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4}$$

n —Number of sensor nodes

k —Number of clusters

l —Message Size

Fig. 4 Ant maze video**Table 1** Simulation parameters

Network parameters	Values
Network Size	100 × 100 m ²
Total Number of nodes	100
Number of Video Nodes	2
Video Sensor Nodes	82 and 91
Routing Protocols	LEACH, LEACH-C, LEACH-GA, GADA-LEACH and μ GA-LEACH
Initial Energy, E _o	0.5, and 1 J/node
Transmitter Energy, E _{TX}	50 nJ/bit
Receiver Energy, E _{RX}	50 nJ/bit
Amplification Energy for short distance, E _{fs}	10 pJ/bit/m ²
Amplification Energy for long distance, E _{mp}	0.0013pJ/bit/m ²
Data Aggregation Energy, E _{da}	5 nJ/bit
Cluster Head probability, p	0.5
Maximum number of Iteration	8000

E_{elec} —Total transmitted energy

E_{DA} —Data aggregation energy

ϵ_{fs} —Short distance amplification energy

ϵ_{mp} —Long distance amplification energy

Table 2 represents the energy consumption of various protocols by transmitting a video over WSN with the initial energy of 1 J/node. Table 3 shows the energy consumption of video transmission with the initial energy of 0.5 (J/node). From the result, we conclude that our proposed protocol consumes lesser energy as 86%,

85%, 52%, and 35% compared with LEACH, Centralized LEACH (LEACH-C), LEACH-Genetic Algorithm (GA), and GADA-LEACH protocol respectively.

Table 2 Evaluation of Energy Consumption with the Initial energy of 1 (J/node)

Frames length	LEACH	Centralized LEACH	LEACH-GA	GADA LEACH	Proposed method
15	14.59	13.1259	4.3881	3.4511	2.0213
30	29.2552	26.4113	9.4292	5.3747	4.0546
45	43.8829	39.6336	14.1437	7.8632	6.0508
60	58.4501	52.7526	17.2923	10.7424	8.0698
75	73.164	65.8368	23.1779	16.9956	10.0999
90	87.7407	79.0793	15.6733	17.4319	12.1146
120	116.9688	105.4299	38.0428	28.6574	16.1557
150	146.1658	131.5856	46.495	27.5437	20.1867
300	292.2663	262.0748	74.3160	73.0487	40.3005
500	486.5904	441.0746	156.4630	103.4266	67.2721
600	583.7459	526.6310	180.6384	138.7263	80.6991
900	876.6371	782.115	269.3112	226.8019	121.4481

Table 3 Evaluation of Energy Consumption with the Initial energy of 0.5 (J/node)

Frames length	LEACH	Centralized LEACH	LEACH-GA	GADA LEACH	Proposed method
15	7.7674	7.4107	4.2161	3.4518	2.0179
30	15.5641	14.8554	8.2345	5.3344	4.0318
45	23.3290	22.2027	10.4722	8.2314	6.0562
60	31.0986	29.6609	16.0468	10.4615	8.0681
75	38.8799	36.9738	20.3924	15.6499	10.1182
90	46.6605	44.3120	23.3571	22.6646	12.1117
120	62.1530	59.2149	32.9584	22.1636	16.1416
150	77.7969	74.1093	26.2696	36.0552	20.1826
300	155.4353	148.4734	52.4030	81.2242	40.4648
500	259.0341	246.7536	132.8491	93.3515	67.2672
600	311.2898	296.5043	106.0136	113.4550	80.8472
750	389.1152	371.5276	207.8653	167.5074	101.0853
900	466.5140	442.9190	238.3538	181.4032	121.2908

7 Conclusion

In this paper, video transmission on hierarchical protocol was presented. Video transmission over the WSN is still a challenging task, and we initiated the new direction for the LEACH based protocol using WSN. Our performance analysis shows that the proposed μ GA-LEACH protocol consumes less energy as 35% to 85% when compared with other LEACH based hierarchical protocols.

Our simulation results are based on single-path routing; in future, it can be implemented on multipath routing. Further, the proposed method can be implemented in real-time applications like surveillance, monitoring, and so on.

References

1. Dong D, Liao X, Liu K, Liu Y, Xu W (2012) Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches. *IEEE Trans Comput* 61(10): 1417–1428. <https://doi.org/10.1109/TC.2011.149>
2. Alberts M et al (2013) New wireless sensor network technology for precision agriculture. In: *Proceedings international conference application information and communication technology (AICT)*, Jelgava, Latvia, pp 25–26
3. Campbell J, Gibbons PB, Nath S, Pillai P, Seshan S, Sukthankar R (2005) IrisNet: an Internet-scale architecture for multimedia sensors. In: *Proceedings 13th ACM international conference multimedia*, pp 81–88
4. Ibrahim A, Han Z, Liu K, Distributed energy-efficient cooperative routing in wireless networks. *IEEE Trans Wirel Commun* 7(10):3930–3941. <https://doi.org/10.1109/T-WC.2008.070502>
5. Radhika M, Sivakumar P (2016) A review on different genetic algorithm for lifetime enhancement of nodes in wireless sensor network. *Asian J Inf Technol* 17(15):3232–3237
6. Radhika M, Sivakumar P (2021) Energy optimized micro genetic algorithm based LEACH protocol for WSN. *Wirel Netw* 27:27–40. <https://doi.org/10.1007/s11276-020-02435-8>
7. Sivakumar P, Radhika M (2018) Performance analysis of LEACH-GA over LEACH and LEACH-C in WSN. *Procedia Comput Sci Elsevier* 125:248–256. <https://doi.org/10.1016/j.procs.2017.12.034>
8. Xiong Z-Y, Fan X-P, Liu S-Q, Zhong Z (2010) Distributed image coding in wireless multimedia sensor networks: A survey. *Proc Adv Comput Intell (IWACI)*, 618–622
9. Radhika M, Sivakumar P (2019) Video traffic analysis over LEACH-GA routing protocol in a WSN. *Procedia Comput Sci Elsevier* 165:701–707. <https://doi.org/10.1016/j.procs.2020.01.028>
10. Jasim A, Ceken C (2015) Video streaming over wireless sensor network. *IEEE conference in wireless sensor*, pp 63–66
11. Ciubotaru B, Pescaru D, Todinca D (2007) Performance analysis on video transmission in a wireless sensor network. *4th international symposium in applied computational intelligence and informatics*, pp 183–186
12. Baguda YS, Faisal N (2008) “Performance analysis of H.264 video over Error-prone” Transmission Environment”. *6th international conference in telecommunication technologies and 2nd Malaysia conference on photonics, proceedings of IEEE*, pp 222–225
13. Ezz El Dien Abd El Kader M, Youssif AAA, Ghalwash AZ (2016) Energy aware and adaptive cross-layer scheme for video transmission over wsns. *IEEE Sens J* 16(21):7792–7802
14. Karimi E, Akbari B (2013) Priority scheduling for multipath video transmission in WMSNS. *Int J Comput Netw Commun* 5(6):167–180

15. Aghdasi HS, Abbaspour M, Moghadam ME, Samei Y (2008) An energy-efficient and high-quality video transmission architecture in wireless video-based sensor networks. *Sensors* 8(8):4529–4559
16. Heinzelman W, Chandrakasan A, Balakrishnan H (2000) LEACH: energy efficient communication protocol for wireless microsensor networks. In: *Proceedings of Hawaii international conference on system science*, Maui, Hawaii, pp 3005–3014
17. Heinzelman W, Chandrakasan A, Balakrishnan H (2002) An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans Wirel Commun* 4(1):660–670. <https://doi.org/10.1109/twc.2002.804190>
18. Liu JL, Ravishankar CV (2011) LEACH-GA: genetic algorithm-based energy efficient adaptive clustering protocol for wireless sensor networks. *Int J Mach Learn Comput* 1(1):79–85. <https://doi.org/10.7763/IJMLC.2011.V1.12>
19. Bhatia T, Kansal S, Goel S, Verma AK (2016) A genetic algorithm based distance-aware routing protocol for wireless sensor networks. *Comput Electr Eng* <https://doi.org/10.1016/j.compeleceng.2016.09.016>
20. https://www.youtube.com/watch?v=YHWZ_wGlc_w
21. Shaikh FK, Zeadally S (2016) Energy harvesting in wireless sensor networks: a comprehensive review. *Renew Sustain Energy Rev* 55:1041–1054. <https://doi.org/10.1016/j.rser.2015.11.010>
22. Anastasi G, Conti M, Di Francesco M, Passarella A (2009) Energy conservation in wireless sensor networks: a survey. *Ad Hoc Netw* 7(3):537–568. <https://doi.org/10.1016/j.adhoc.2008.06.003>
23. Jung JW, Weitnauer M (2013) On using cooperative routing for lifetime optimization of multi-hop wireless sensor networks: analysis and guidelines. *IEEE Trans Commun* 61(8):3413–3423. <https://doi.org/10.1109/TCOMM.2013.052013.120707>
24. Chidean MI, Morgado E, Sanroma'n-Junquera M, RamiroBargueno J, Ramos J, Caamano AJ (2016) Energy efficiency and quality of data reconstruction through data-coupled clustering for self-organized large-scale WSNs. *IEEE Sens J* 12(16):5010–5020. <https://doi.org/10.1109/jsen.2016.2551466>
25. Peiravi A, Mashhadi HR, Javadi SH (2013) An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm. *Int J Commun Syst* 1(26):114–126. <https://doi.org/10.1002/dac.1336>
26. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. *Comput Netw* 12(52):2292–2330. <https://doi.org/10.1016/j.comnet.2008.04.002>
27. Karahan A, Erturk I, Atmaca S, Cakici S (2014) Effects of transmit-based and receive-based slot allocation strategies on energy efficiency in WSN MACs. *Ad Hoc Netw* 13:404–413. <https://doi.org/10.1016/j.adhoc.2013.09.001>
28. Demigha O, Hidouci WK, Ahmed T (2013) On energy efficiency in collaborative target tracking in wireless sensor network: a review. *IEEE Commun Surv Tutor* 15(3):1210–1222. <https://doi.org/10.1109/SURV.2012.042512.00030>
29. Rawat P, Singh KD, Chaouchi H, Bonnin JM (2014) Wireless sensor networks: a survey on recent developments and potential synergies. *J Supercomput* 68(1):1–48. <https://doi.org/10.1007/s11227-013-1021-953>
30. Basaran C, Kang KD (2009) Quality of service in wireless sensor networks. In: *Guide to wireless sensor networks*, pp 305–321. https://doi.org/10.1007/978-1-84882-218-4_12
31. Zhai C, Liu J, Zheng L, Xu H, Chen H (2012) Maximise lifetime of wireless sensor networks via a distributed cooperative routing algorithm. *Trans Emerg Telecommun Technol* 23(5):414–428. <https://doi.org/10.1002/ett.2498>

Hybridization of Texture Features for Identification of Bi-Lingual Scripts from Camera Images at Wordlevel



Satishkumar Mallappa, B. V. Dhandra, and Gururaj Mukarambi

Abstract In this paper, hybrid texture features are proposed for identification of scripts of bi-lingual camera images for a combination of 10 Indian scripts with Roman scripts. Initially, the input gray-scale picture is changed over into an LBP image, then GLCM and HOG features are extracted from the LBP image named as LBGLCM and LBHOG. These two feature sets are combined to form a potential feature set and are submitted to KNN and SVM classifiers for identification of scripts from the bilingual camera images. In all 77,000-word images from 11 scripts each contributing 7000-word images. The experimental results have shown the identification accuracy as 71.83 and 71.62% for LBGLCM, 79.21 and 91.09% for LBHOG, and 84.48 and 95.59% for combined features called CF, respectively for KNN and SVM.

Keywords LBP · LBGLCM · LBHOG · GLCM · HOG · KNN · SVM

1 Introduction

In the present situation there is a significant improvement in internal storage, the performance of the processors, and high resolution handheld mobile devices such as smartphones; have emerged new ways of capturing and processing images in general and document images in particular. With these handheld mobile devices, users can easily capture the images of hard documents, too fragile documents, text present in the scene, signboards, text on monuments, business boards, digital boards, historical documents, text written on leaves, carved on stones and so on. Along with

S. Mallappa (✉)

Department of P.G.Studies and Research in Computer Science, Gulbarga University, Gulbarga, India

e-mail: satishkumar697.compssc@gug.ac.in

B. V. Dhandra

Department of Statistics, Christ (Deemed to Be University), Bengaluru, India

G. Mukarambi

School of Computer Science, Department of Computer Science, Central University of Karnataka, Gulbarga, India

these advantages, there is a possible presence of distortions like uneven illumination, blur, out-of-focus, perspective distortion etc. called noises. Processing such camera captured document images are challenging problems. These challenges have motivated us to work on these problems. Hence, the identification of script from the bilingual document images captured from the camera at the word level by using mixture of features of textures are addressed in this paper. In Sect. 2, review of literature is presented, details of the proposed method and data set considered are addressed in Sects. 3, and 4 contains exploratory outcomes and discussion, and Sect. 5 contains the conclusion.

2 Review of Literature

The details of research work found in relation with the camera captured document image script identification is discussed in following.

Ojala et al. [1] they have mentioned the SFTA features to identify bi-script from camera based multi-lingual images containing text, by considering the combination of 11 language texts viz., Kannada, Bangla, English, Tamil, Oriya, Telugu, Hindi, Gujarati, Malayalam, Urdu, and Punjabi and used Feed Forward Feature selection technique they got the recognition rate of 87.02% from English and Hindi script combinations by KNN classifier.

Dalal and Triggs [2], they reported the bus signboard script identification using the Log-Gabor, Gabor, and wavelet features to identify the Malayalam Kannada, and English scripts. In segmentation, they adopted a set of morphological operations to extract the text from the bus signboard image. From that they constructed the (Gabor = 16, Log-Gabor = 16 and Wavelet = 28) and they have selected the 60 texture features for each word. To classify the scripts, they have used the k-NN and SVM classifiers, and obtained 97.40% recognition accuracy.

Li and Tan [3], have proposed the signature and template similarity measures to recognize the scripts of the camera-based documents containing Cyrillic, Roman, Greek, Thai, Hebrew, Arabic, Bengali, Japanese, and Chinese. With the cosine, they have computed the Hamming distance between the components of query document and template, and submitted it to NN classifier. With these, they could achieve 91.00% of recognition accuracy.

Dhendra et al. [4], reported SFTA features to identify the scripts at the block level by using the two-language document captured from the camera. And they got 86.45% recognition accuracy in connection with English and Gujarati script combinations using the KNN classifier.

Mukarambi et al. [5], they reported features obtained by using LBP method is utilized for the purpose to identify document image containing three scripts in it. At the block-wise having Hindi, Kannada, and English scripts. From the LBP method they obtained the 59 features and fed them to the KNN and SVM classifiers and got the maximum recognition accuracy as 99.70, and 98.00%, from KNN and SVM classifiers.

Malik et al. [6], reported work on script identification of Chinese, Arabic, Hindi, Urdu, and English script video images. They have extracted and selected the combination of Gabor, GLCM, LBP, HOG, SFTA, LPQ, and AR Coefficients textural features as their potential feature set. For identification, they have used ANN classifier and achieved 86.54% accuracy for a single LPQ feature and 96.75% accuracy for combined features.

Shivakumar et al. [7], has done the work on script identification from videos containing Chinese, English, Japanese, Arabic, Tamil, and Korean scripts at the word level, using Gradient-Angular-Features (GAF) as potential features. The gradient angle segmentation process has been used to segment the word from video frames. For the experiment, they have created the template for each script, and GAF features are extracted from each template and achieved the **88.2%** average recognition accuracy and used the Euclidean distance between templates of the script with the labeled script as the similarity measure.

Dhendra et al. [8], have carried out work on eleven Indian scripts. The extracted features from SFTA, LBP, and mixture of features were to identify the scripts from the document captured from the camera. The 18 features were extracted from the SFTA and LBP method extracted 59 features and concatenated the SFTA and LBP features. From the extracted features they got maximum of 77.94 and 82.39% by using SFTA features and 89.82 and 93.94% by using LBP features, from SVM and KNN classifiers for a mixture of features KNN has given 94.45%, and SVM has shown 93.88% recognition accuracy.

Dhendra et al. [9], have presented the work on two and three types of script identification from camera-captured document images at line-level using SFTA and LBP features. For the experiment, they have taken 11 scripts. The final recognition rate for bi-script is 98.78% and for tri-script 97.66% is obtained from SVM classifier.

3 Proposed Method

This part of the paper gives the details about extracting the potential features for identification of the scripts from the bilingual document image captured from the camera is proposed. Initially, the LBP image is generated from the input image to have normalized and more texture information. From the LBP image, second-order statistical features (GLCM) which contain the local features named as Local Binary Gray Level Co-occurrence Matrix (LBGLCM) [10, 11], and HOG features called as Local Binary Histogram Oriented Gradient (LBHOG) features are extracted, then both features are combined to form a potential feature vector for identification of scripts of the camera based bilingual document image. The experiments are carried out for individual and combined features to study their performances (see Fig. 1).

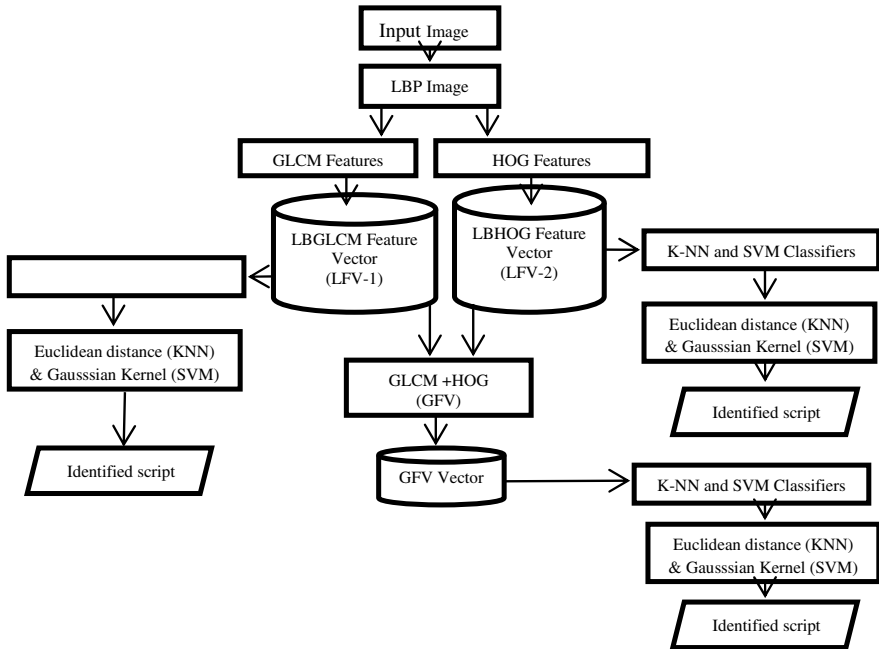


Fig. 1 Block diagram of the proposed approach

3.1 Creation of LBP Image

To obtain the texture descriptor, the LBP operator has been applied to gray-scale images. The LBP operator transforms the gray-scale images into integer labels, which are the description of small-scale appearance in the image. The image labels are used for further analysis of an image. The LBP operator creates a mask of size 3×3 , and it slides on all over the image. In each slide, it creates the LBP patterns by making the comparison with central pixel value with its eight neighborhood pixels values. If neighborhood pixel value is greater than or equal to center pixel value, then it assigns 1 otherwise 0. The following equation exhibits the central pixel value calculation.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} S(g_p - g_c)2^p \tag{1}$$

$$\text{where } S(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

g_c is the gray value of the central pixel, g_i is the gray value of the neighboring pixel and P is the number of neighbors, and R represents the radius of the neighborhood.

The following Fig. 2 represents the sample of generating the LBP image from the Input gray-scale image (see Fig. 3).

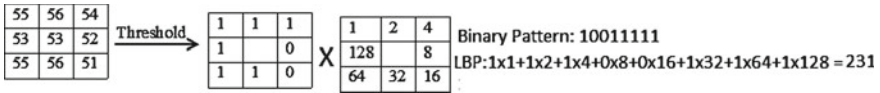


Fig. 2 Local binary pattern computation

Script	Gray-scale image	LBP Image	Script	Gray-scale image	LBP Image
English			Punjabi		
Hindi			Tamil		
Bangla			Kannada		
Oriya			Telugu		
Malayalam			Urdu		
Gujarati					

Fig. 3 Sample of the input gray-scale image and their corresponding LBP image

3.2 Extraction of GLCM Feature from LBP Image

This feature is popular and much used to evaluate the textures of the image. It enhances the texture information of the image and gives a straightforward interpretation. This tabulates occurrences of Gray scale levels of an image. Typically, the GLCM features extracted from an input image based on the pixel and its corresponding neighbor pixels ignoring the spatial relationship between the local texture patterns of the image. The GLCM features extracted from the LBP image based on the local neighborhood pixels contain the spatial and texture features information with its different patterns, Sastry et al. [11]. In the proposed method following five properties of GLCM are extracted.

(i) Contrast: This parameter reveals the intensity contrast values between the different textural patterns. The contrast is given by.

$$\text{Contrast} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} |i - j|^2 \text{LBP}(i, j) \tag{3}$$

where LBP is the image generated from a gray-scale image.

(ii) Homogeneity: It computes the closeness of gray values distribution of the LBP image. The following equation is used to compute the homogeneity of the texture values from the LBP images.

$$\text{Homogeneity} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\text{LBP}(i, j)}{1 + |i - j|} \quad (4)$$

(iii) Correlation: Correlation is used to compute the degree of association between patterns of textures at the specified pixel position which are relative to each other. The following equation is used to compute the correlation of texture patterns in the image.

$$\text{Correlation} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{\{i \times j\} \times \text{LBP}(i, j) - \{\mu_x \times \mu_y\}}{\sigma_x \times \sigma_y} \quad (5)$$

where μ_x, μ_y are the means and σ_x, σ_y are the standard deviations of the probability matrix P_x and P_y . G is the number of Gray levels used.

(iv) Energy: The energy feature computes uniformity of textural pixels. The maximum energy will occur when texture or image-level distribution is either constant or uniform periodically. Following equation describes the energy of the textures in the LBP image.

$$\text{Energy} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (\text{LBP}(i, j))^2 \quad (6)$$

(v) Entropy: This measures the degree of the disorder among the image pixels. It is an inverted portion that has correlated with uniformity. The image which has the larger number of gray levels has larger entropy. The following formula gives the entropy of a gray-scale image.

$$\text{Entropy} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \text{LBP}(i, j) \log(i, j) \quad (7)$$

GLCM has been constructed to get statistical texture features from the second-order features; Contrast, Homogeneity, Correlation, Energy, and Entropy.

Algorithm 1 is designed to extract GLCM features from the LBP images of the input documents for identification of the script of the document using the KNN classifier.

ALGORITHM 1 Extraction of GLCM features from bilingual input document images and Script identification.

Input Image: Word-level LBP images.

Output: GLCM Features and Identified script of the bilingual document image.

Training Phase:

Start

Step 1: Read LBP image generated from the input image

Step 2: Compute the Contrast of an LBP image

Step 3: Compute the Homogeneity of an LBP image

Step 4: Compute the Correlation of an LBP image

Step 5: Compute the Energy of an LBP image

Step 6: Compute the Entropy of an LBP image

Step 7: Four features are generated from Steps 2 to 6 with an angles (0° , 45° , 90° and 135°) of and total makes $4 \times 5 = 20$ features.

Step 8: Save features into the knowledge base along with label.

Stop.

Testing Phase:

Start

Step 1: Repeat steps 1 to 7 from training.

Step 2: Save the features in test database.

Step 3: Calculate the Euclidean distance in between test features and train features.

Step 4: Submit the distance metric to KNN and SVM Classifier to identify the script of the input document.

Step 5: Gives the identified script

Stop.

3.3 Extraction HOG Feature from LBP Image

Dalal and Triggs [2], have originally done the work on HOG feature extraction procedure. This method divides the input image into small patches known as cells. From each cell, a Histogram oriented Gradient features are extracted. HOG is a popular method for its ability to capture the edges and corner details of the input image.

LBP features will not consider the edge, occurrence of intensity, and corner of the shape, whereas HOG captures the details left by the LBP method. Hence, to overcome from the limitation of LBP, the HoG features were considered. The recognition accuracy was enhanced, by using HoG features extracted from the input image.

Algorithm: Script identification from HOG features.

Input : LBP image

Output: Identified script.

Training:

Start

Step 1: Read the LBP image generated from the input image

Step 2: Apply 3 x 3 sliding window

Step 3: Use filter kernel $[-1,0,1]$ and $[-1, 0, 1]$

Step 4: Calculate Histogram of Oriented Gradient direction of an LBP image

Step 5: Generate the Histogram of all cells

Step 6: Normalize the Histograms by using an L2 Normalization method

Step 7: Generate 81 features

Step 8: Store the 81 HOG feature vector into the knowledge base with labels.

Stop.

Testing:

Start

Step 1: Repeat 1 to 7 steps from above train phase.

Step 2: Save features in test database.

Step 3: Calculate the Euclidean metric in between test and train features saved in knowledge base.

Step 3: Submit the distance metric obtained in Step3 to KNN and SVM Classifier to identify the script.

Stop.

3.4 Combined Feature Vector of GLCM and HOG from LBP Image

In this section, the above generated two feature vectors LBGLCM and LBHOG are combined to form the Global Feature Vector (GFV). This GFV contains the GLCM and HOG features derived from the LBP image.

Algorithm: Identification Script from the LBP image with GFV.

Input: LBGLCM and HOG features

Output: Identified Script.

Training phase:

Start

Step 1: Load the LBGLCM

Step 2: Load the LBHOG

Step 3: Combine the LBGLCM and LBHOG to form $GFV = [LFV-1, LFV-2]$

Step 4: Store the GFV in knowledge base with script labels.

Stop

Testing Phase:

Start

Step 1: Repeat the steps 1 to 4 from above train phase

Step 2: Save GFV features in the test base

Step 3: Calculate the distance metric in between test and train features and submit to KNN and SVM.

Step 4: Display the identified script.

Stop.

4 Experimental Results and Discussion

The experiment for the proposed methods are carried out with self created dataset of 77,000 images containing 11 scripts collected from newspapers, Magazines, Novels, Computer Printouts, etc. and each script has a 10,000 words image, as of now there is no open access data set.

4.1 Results and Discussion

In the following tables presents the recognition accuracy of bi-scripts with the said feature set submitted to SVM and KNN classifiers. For the KNN classifier correlation distance metric is used for similarity measure with $K = 3$ as the optimal value. In the SVM classifier, the Gaussian kernel is used.

From the above Table 1, is shown the LBGLCM features came out as the important features to recognize the English with Hindi scripts by attaining 90.03% and 90.32% recognition accuracies from KNN and SVM classifiers, respectively. The average recognition accuracies are 71.83, and 71.62% for KNN and SVM classifiers and it needs a further increase in recognition accuracies.

The observation from Table 1 reveals that the combination of English and Hindi has achieved the highest recognition accuracy of 87.47 and 96.77% for KNN and

Table 1 Average recognition accuracy of LBGLCM and LBHOG features with KNN and SVM Classifiers

Bi-lingual script identification based LBGLCM 20 features				Bi-lingual script identification based LBHOG (LFV-2) 81 features	
Sl. No	Scripts	KNN (%)	SVM (%)	KNN (%)	SVM (%)
1	E-H	90.03	90.32	87.47	96.77
2	E-B	64.77	62.61	85.85	92.31
3	E-G	71.37	69.26	72.21	88.48
4	E-P	69.04	65.39	85.15	94.43
5	E-O	86.35	90.21	78.80	95.02
6	E-U	71.66	75.92	82.05	92.19
7	E-K	66.24	65.89	79.88	90.82
8	E-Te	63.06	61.65	70.86	85.69
9	E-Ta	70.81	74.74	77.84	90.19
10	E-M	65.00	60.25	72.05	85.04
Avg. Rec. Acc		71.83	71.62	79.22	91.09

Note U- Urdu, E-English, B-Bangla, P-Punjabi, Te-Telugu, H-Hindi, O-Oriya, M-Malayalam, G-Gujarathi, K-Kannada, Ta-Tamil

SVM classifiers. The average recognition accuracies for KNN and SVM classifiers are 79.22%, and 91.09% respectively. The comparison between the LBGLCM and LBHoG features is that LBHoG features perform much better than LBGLCM features (Tables 1 and 2). Table 2 shows the performance of the LBGLCM and LBHoG (GFV) combined features with KNN and SVM classifiers.

Table 2 The total recognition accuracy of combined features (GFV)

Bi-lingual script identification based GFV 101 features			
Sl. No.	Scripts	KNN (%)	SVM (%)
1	E-H	96.86	99.56
2	E-B	88.06	98.09
3	E-G	79.69	95.59
4	E-P	87.89	96.70
5	E-O	89.63	94.44
6	E-U	84.78	98.05
7	E-K	84.58	96.62
8	E-T	73.38	95.65
9	E-T	83.16	90.63
10	E-M	76.77	90.63
Avg. Rec. Acc		84.48	95.60

Fig. 4 The comparative recognition accuracy of the feature vectors using the KNN classifier

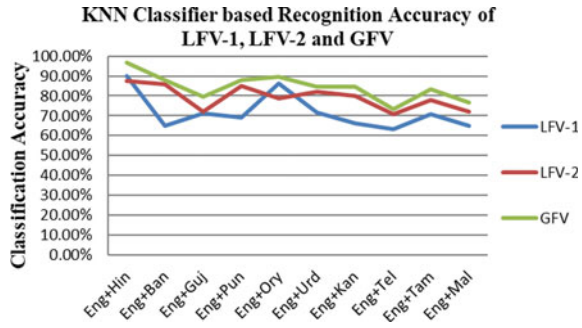
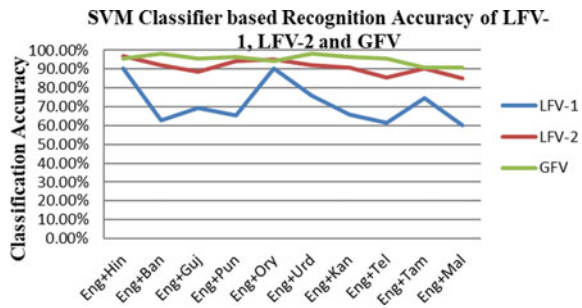


Fig. 5 The comparative recognition accuracy of the feature vectors using the SVM classifier



From the above Table 2, it is clear that the GFV features are potential to recognize the English and Hindi scripts, achieving maximum recognition accuracy of 96.86% and 99.56% with KNN and SVM classifiers, respectively. The average recognition accuracy of the 10 scripts with English is 84.48%, for KNN and 95.60% for SVM classifiers. Hence, the proposed combined features GFV outperforms for all 10 bilingual scripts. However, still there is the possibility of reducing the feature size by applying PCA, which will be considered in the next phase (see Figs. 4 and 5).

From the above tables and figures, it is clear that GFV features with SVM classifiers outperformed as compared to both the LBGLCM and LBHoG features.

5 Conclusion

This proposed method first creates an LBP image derived from the grayscale input image. From LBP image the HoG and GLCM features are extracted and it is called LBGLCM and LBHoG features and these features are concatenated to form a single feature vector called GFV. Algorithms are designed to carry out the performance evaluation of the proposed three feature vectors with respect to recognition accuracies. All these feature vectors are submitted to SVM and KNN classifiers. The combined feature vector GFV has outperformed compared to LBGLCM and LBHoG feature vectors. The average recognition accuracy of 95.59% is obtained with SVM classifiers.

References

1. Ojala T, Pietikainen M, Harwood D (1996) A comparative study of texture measures with classification based on feature distributions. *Pattern Recogn* 29:1996
2. Dalal N, Triggs B (2002) Histograms of oriented gradients for human detection. In: *IEEE conference on computer vision and pattern recognition*, 2005
3. Li L, Tan CL (2008) Script identification of camera-based images. *IEEE*. 978-1-4244-2175-6/08/2008
4. Dhandra BV, Mallappa S, Mukarambi G (2019) Script identification of camera-based bilingual document images using SFTA features. *IJTHI* 1–12
5. Mukarambi G, Mallappa S, Dhandra BV (2017) Script identification from camera-based Tri-Lingual document. *IEEE*, pp 214–217
6. Malik Z, Mirza A, Bennour A, Siddiqi I, Djeddi C (2015) Video Script Identification using combination of textural features. *IEEE*, pp 61–67
7. Shivakumara P, Sharma N, Pal U, Blumenstein M, Tan CL (2014) Gradient-angular-features for word-wise video script identification. In: *22nd international conference on pattern recognition*. *IEEE*, pp 3098–3103
8. Dhandra BV, Mallappa S, Mukarambi G (2020) Camera-based tri-lingual script identification at word level using a combination of SFTA and LBP features. *Int J Adv Sci Technol* 29(3):6609–6617
9. Dhandra BV, Mallappa S, Mukarambi G (2020) Script identification at Line-level using SFTA and LBP features from Bi-lingual and Tri-lingual documents captured from the camera 13(4):975–980
10. Öztürka Ş, Akdemir B (2018) Application of feature extraction and classification methods for histopathological image using GLCM, LBP, LBGLCM, GLRLM and SFTA. *ICCIDS-2018*
11. Sastry SS, Kumari TV, Rao CN, Mallika K, Lakshminarayana S, Tiong HS (2012) Transition temperatures of thermotropic liquid crystals from the local binary gray level co-occurrence matrix. *Adv Condens Matter Phys*

Advanced Algorithmic Techniques for Topic Prediction and Recommendation—An Analysis



Shaik Nazeer , Prathyusha Yayavaram , and L. Jani Anbarasi 

Abstract With the rapid growth in technology and the need for people to stay connected, social networking has become the most essential part of day-to-day life. As the demand went high, a lot of social networking applications were made available to the users. Twitter is one such famous platform where users get to post short and crisp texts called tweets. Being restricted with a limit of 140 characters per tweet is one of the effective ways to diffuse information. Twitter has a unique ability to organize and encourage large-scale debates about ‘trends.’ Users can stay updated with the latest trends around the world by using twitter’s large-scale platform of posting tweets, which are updated in real-time. Twitter’s uniqueness from other social media platforms comes from its ability to list out hot trending topics being discussed by a large number of people. Predicting the topics which are soon be trending is a significant way to help users who want to grow their following by communicating their message to others. In this paper, a recommendation based on pre-trending subjects based on individual user interests, as well as prediction algorithms for identifying tweets that will shortly trend has been detailed. Techniques like Hashtag, Word Ranking, Authority, Background Tweet Detection and Short-Term Fluctuation Modeling based Approaches are analyzed to show the performance of predictions.

Keywords Tweet prediction · Hash tag · Word ranking · Shortest term fluctuation model · Background Tweet detection

S. Nazeer
VMware Pvt. Ltd., Bengaluru, India
e-mail: shaik.nazeer2015@vit.ac.in

P. Yayavaram
Myntra Pvt. Ltd., Bengaluru, India
e-mail: yayavaramsai.prathyusha2015@vit.ac.in

L. J. Anbarasi (✉)
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
e-mail: janianbarasi.l@vit.ac.in

1 Introduction

Social media has become an essential part of our day-to-day life with its capability of connecting people and providing a platform where people can learn about the trending topics across the world. Twitter is one such platform which helps people follow their favorite celebrities, entrepreneurs, political leaders etc. all over the world. Zubiaga et al. [1] evaluated the tweet trends pattern among the users of social media with shared interests. His work included the analysis on the subjects that could trigger a trend in Twitter and categorized the types of trends into four types: news, current affairs, comic content and commemoratives. The goal is to predict and categorize the trending topics earlier than when they happen so that they can deliver a selective subset of trends to the end users matching their interests. Tweet relevancy to the user is the biggest concern while displaying tweets from such a huge collection of tweets. To overcome such a problem, relevancy engines were introduced which categorize the tweets based on user's personal data like geographic location, his/her interests, previous tweet shares made by the user and etc. Kywe et al. [2] has introduced a hashtag recommendation system that helps in categorizing the targeted users to improve the relevance of the tweets they see. But hashtags do not have any centralized organization and the representations of hashtags made are not controlled. A given hashtag can have multiple meanings in different geographical areas. The recommendation that solely depends on the hashtags would not yield results with high accuracy. The algorithm for hashtag-based recommendation systems has been evaluated and results are compared against other recommendation systems. The improvements and more sophisticated approaches to relevancy engines will help in accurate tweet suggestions. Predicting the soon trending tweets is yet another challenging area that helps users to get relevant tweets with high accuracy. Soon trending tweet detection algorithms based on a variety of parameters are being proposed in this paper. Go et al. [3] performed a twitter sentiment analysis and Waters et al. [4] performed a content-based analysis for a non-profitable organization. Borgmann et al. [5] performed a qualitative analysis of participants along with their tweet strategies and contents for the urological field. Pemmaraju et al. [6] analyzed the best practices and uses of tweets for the health care sector. Duan et al. [7] proposed a ranking model that not only used the contents based on relevance but also the authority and the URL of the link.

This paper has been organized as follows. Section 2 focuses on the related work on methodologies performed in forecast Tweets. Section 3 explains the results and discussion part which deals with the predictions and accuracy of the proposed model. Section 4 details the conclusion and future scope of this research.

2 Proposed Model

The research’s motive is to validate various algorithms that could anticipate which subjects and hashtags will become popular soon. To do so, five parallel systems are created, each taking a different approach to the problem. Since trends are not necessarily topics that are frequently discussed, but topics that are rapidly increasing in popularity or temporarily increasing in usage, an effective way of predicting trends could be tracking word usage in tweets based on baseline or normal usage. Each algorithm collected a corpus of tweets using Twitter’s Streaming API service [8] and then delivered the top 50 topics that are “predicted” to be soon trending. These predicted topics were then compared to the top Twitter topics that are retrieved during the following few hours. The architecture of the system is shown in Fig. 1.

2.1 Hashtag-Based Approach

Hashtag-based Approach [9] is a straightforward method where the frequency of each hashtag is calculated from the corpus of tweets and these hashtags are ranked based on their frequency. Hashtags are often chosen with care because they symbolize the message that people are attempting to convey. Twitter Streaming API has been used to download the tweets which are used to count the frequency of each hashtag. Following the collection of tweets, a tag dictionary is created to track the number of times each hashtag has been used.

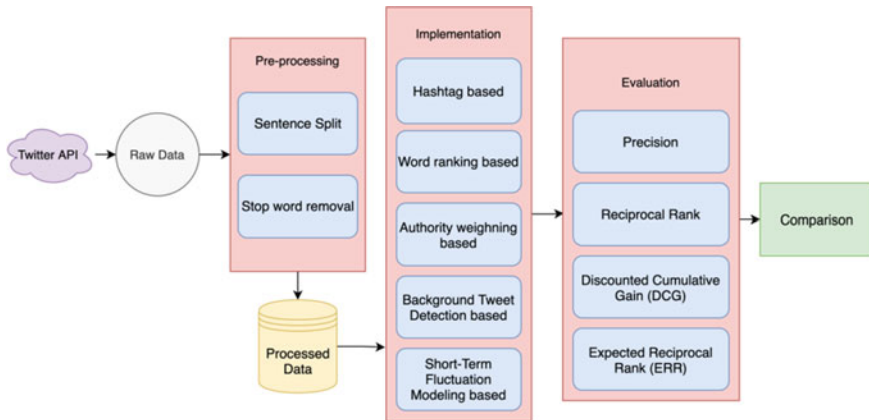


Fig. 1 Architecture of the proposed system

2.2 Word Ranking Based Approach

Word Ranking based approach ranks the topics based on the most frequently observed words. This technique is similar to the Hashtag method in terms of calculating the number of times a topic appears in a tweet, but instead of hashtags, the words of a tweet's text are considered. Additional refining of data was required as moved from hashtags to words, which included irrelevant data like grammatical constructors, unseemly words, stop words [10], etc. After pruning the unwanted data, this approach provided more meaningful data which resulted in better predictions.

2.3 Authority Weighting Based Approach

This method distributes weights [11] to terms to achieve more accurate predictions. Weights are assigned based on the popularity of the users, their profile verification status, number of followers, and their previous trend history. These kinds of weights assigned to tweets help in modeling the data more accurately. Considering the previous trend history of the users helps draw more information on predicting the popularity of their future tweets. Unlike the previous two methods, users' characteristics are taken into account along with tweet words segregation which would help in more meaningful predictions. The algorithm for Word Ranking Based approach is shown as follows. The computation of weights for accurate predictions is performed using the following Eqs. (1–5).

$$\text{Rank By : } (K_1 \times \text{freq}_{x_1\text{mod}} + K_2 \times \text{follow}_{x_2\text{mod}} + K_3 \times \text{ver}_{x_3\text{mod}})\varphi \quad (1)$$

K_1, K_2, K_3 are arbitrary weights.

$$\text{freq}_{x_y\text{mod}} = \frac{\log \log(\log \log(\text{freq}_x)) - \mu_{\log \log(\log \log(\text{freq}))}}{\sigma_{\log(\log \log(\text{freq}))}} \quad (2)$$

$$\text{follow}_{x_y\text{mod}} = \frac{\log \log(\text{follow}_x) - \mu_{\log \log(\text{follow})}}{\sigma_{\log(\text{follow})}} \quad (3)$$

$$\text{ver}_{x_y\text{mod}} = \frac{\log \log(\text{ver}_x) - \mu_{\log \log(\text{ver})}}{\sigma_{\log(\text{ver})}} \quad (4)$$

$$\varphi = \{4 \text{ if term present in hashtag 1 otherwise} \quad (5)$$

The histogram of word frequencies, log log word verified user frequencies, and log verified frequencies are computed based on the number of tweets, number of verified users per word, and frequency of words and is shown in Fig. 2.

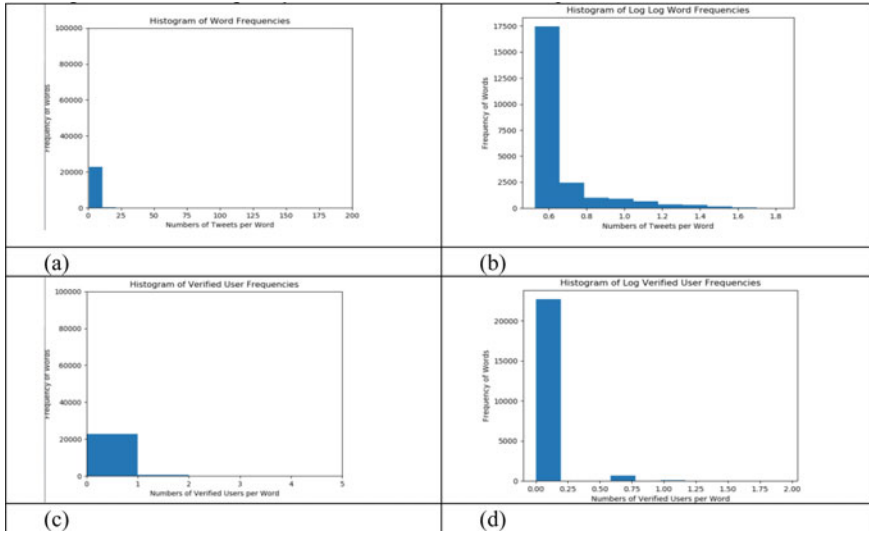


Fig. 2 Histogram distribution, **a** Word frequency, **b** Log Word Frequency, **c** Verified user frequency, **d** Log verified user frequency

2.4 Background Tweet Detection-Based Approach

Using historical tweet data [12], this method identifies subjects that are referenced at unusually high rates. It can be noticed that trends aren't necessarily issues that are addressed infrequently. There is a significant number of tweets that trend because of other factors like transient spikes in usage, rapidly gaining popularity, etc. Hence the frequency of words used in a tweet can be compared to their standard or typical usage which will predict more accurate results. To achieve this, the frequency of tweets in the test sample is compared to the corresponding frequencies in the historical data. The algorithm for the Background Tweet Detection approach is shown below. The tweets are ranked using Eq. (6).

$$\text{Rank By : } \left(\frac{\text{freq}_{\text{curr}} - \text{freq}_{\text{hist}}}{\text{freq}_{\text{hist}}} \right)^3 \tag{6}$$

2.5 Short-Term Fluctuation Modeling Based Approach

Not only the frequencies and occurrence of the individual words [13], but tweet subjects also play an important role in predicting the tweet trends. More information on tweets subjects can be obtained by comparing the variance in the usage of each

topic over a period, which gives growth outlooks and predicts if they would soon be trending. This can be obtained by computing term frequencies separately for chunks of different periods of time and determining the upward or downward frequency movement from their differences. The algorithm for the Short-Term Fluctuation Modeling approach is shown below. The rank of the tweets is computed as given in Eq. (7).

$$\text{Rank By : } (K_1\Delta_{5-4} + K_2\Delta_{4-3} + K_3\Delta_{3-2} + K_4\Delta_{2-1}) \quad (7)$$

3 Results and Discussion

To assess the accuracy of the predictions, a set of assessment metrics has been used to compare and analyze the outcomes of the algorithms with the actual trended topics. Over 50 predicted hashtags and topics of tweets are used for the analysis to correlate with the actual tweets that started to trend in the following hours of the predictions. The results have been categorized into completely relevant, slightly relevant, and completely irrelevant to the actual trending tweets and is given in Table 1.

The precision [14, 15] is computed using Eq. (8).

$$P = \text{relevant tweets retrieved} / \text{Relevant tweets} \quad (8)$$

The reciprocal rank (RR) measures the possible sample tweets ordered with the correctness probability where k refers to the sample tweet and rank prefers the best for the query as shown in Eq. (9).

$$\text{Reciprocal Rank} = \frac{1}{k} \sum_{j=1}^k \frac{1}{\text{rank}_j} \quad (9)$$

Discounted Cumulative Gain (DCG) refers to the relevance values from all the generated results as shown in Eq. (10) where rk_j shows the relevance in position j .

Table 1 Table captions should be placed above the tables

	Hashtag	Word ranking	Authority weighting	Background tweet detection	Short-term fluctuation modeling
Precision	0.4	0.2	0.42	0.5	0.44
RR	0.5	0.25	1	0.33	1
DCG	14.73	5.09	12.26	20.38	20.1
ERR	0.81	0.09	0.18	0.57	1.46

$$DCG = \sum_{j=1}^k rk_j \quad (10)$$

Expected Reciprocal Rank (ERR) determines the expected time expected for the user to retrieve the relevant tweet as given in Eq. (11).

$$ERR(j) = \sum_{j=1}^k \left(\frac{1}{j}\right) * \prod_{k=1}^{j-1} (1 - S(Sc_k)) * S(Sc_j) \quad (11)$$

With the highest score in Discounted Cumulative Gain (DCG) and Precision, Background Tweet Detection-based Approach implies to be the most efficient prediction algorithm with the most relevant predicted tweet topics. However, the Reciprocal rank and Expected Reciprocal Rank, 0.33 and 0.57 respectively, are lower than the majority of the other techniques, indicating that its ranking order is not so accurate despite being highly accurate overall. Short-Term Fluctuation Modeling based Approach, on the other hand, has the highest Expected Reciprocal Rank (ERR) of 1.46 and second highest Precision and Discount Cumulative Gain, suggesting it to be the most balanced and accurate algorithm among all five.

4 Conclusion

In this research work, many algorithms that may predict the twitter topics that will be trending soon and provide users with better recommendations are investigated. Background Tweet Detection and Short-Term Fluctuation Modeling are found to be the most effective after testing each of the prediction methods. The Hashtag, Work Ranking, and Authority Weighing focused on monitoring term frequencies. But other two methods took into account a broader range of frequency patterns throughout time and analyzed a variety of regular frequency trends to better discover anomalous frequency patterns. Evidently, Background Tweet Detection is the most precise and effective technique of prediction analysis, yielding the most forecasts that ultimately trended. In terms of precision, Short-Term Fluctuation Modeling came up second but greatly outperformed Background Tweet Detection in terms of reciprocal rank, implying an improved ranking system that positions the most pertinent recommendations on the top. In addition, it was able to provide value for all of our test subjects by leveraging previous user tweet data to uncover future areas of interest. These techniques and algorithms can be improved in the future to better assist people in utilizing Twitter's great tool. Future work includes the prediction of Tweets based on daily Trending where the relevance can be identified based on the pre-trending hashtags. Similarly, machine learning algorithms can be performed to predict the trends.

References

1. Zubiaga A, Spina D, Martínez R, Fresno V (2015) Real-time classification of twitter trends. *J Am Soc Inf Sci* 66(3):462–473
2. Kywe SM, Hoang TA, Lim EP, Zhu F (Dec 2012) On recommending hashtags in twitter networks. In: *International conference on social informatics*. Springer, Berlin, Heidelberg, pp 337–350
3. Go A, Huang L, Bhayani R (2009) Twitter sentiment analysis. *Entropy* 17:252
4. Waters RD, Jamal JY (2011) Tweet, tweet, tweet: a content analysis of nonprofit organizations' Twitter updates. *Publ Relat Rev* 37(3):321–324
5. Borgmann H, Woelm JH, Merseburger A, Nestler T, Salem J, Brandt MP et al (2016) Qualitative Twitter analysis of participants, tweet strategies, and tweet content at a major urologic conference. *Can Urol Assoc J* 10(1–2):39
6. Pemmaraju N, Mesa RA, Majhail NS, Thompson MA (Oct 2017). The use and impact of Twitter at medical conferences: best practices and Twitter etiquette. In: *Seminars in hematology*, vol 54, no 4. WB Saunders, pp 184–188
7. Duan Y, Jiang L, Qin T, Zhou M, Shum HY (Aug 2010) An empirical study on learning to rank of tweets. In: *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, pp 295–303
8. Twitter API (2010). <http://apiwiki.twitter.com/>
9. Tran VC, Hwang D, Nguyen NT (2018) Hashtag recommendation approach based on content and user characteristics. *Cybern Syst* 49(5–6):368–383
10. Dolamic L, Savoy J (2010) When stopword lists make the difference. *J Am Soc Inform Sci Technol* 61(1):200–203
11. Zhai Y, Li X, Chen J, Fan X, Cheung WK (Sept 2014) A novel topical authority-based microblog ranking. In: *Asia-Pacific web conference*. Springer, Cham, pp 105–116
12. Alessa A, Faezipour M (2019) Preliminary flu outbreak prediction using twitter posts classification and linear regression with historical centers for disease control and prevention reports: prediction framework study. *JMIR Publ Health Surveill* 5(2):e12383
13. Jónsdóttir GM, Milano F (2019) Modeling of short-term tidal power fluctuations. *IEEE Trans Sustain Energy* 11(4):2337–2344
14. Prabhakar N, Anbarasi LJ (2021) Exploration of the global air transport network using social network analysis. *Soc Netw Anal Min* 11:26. <https://doi.org/10.1007/s13278-021-00735-1>
15. Sharon JJ, Jani Anbarasi L, Raj BE (2018) DPSO-FCM based segmentation and classification of DCM and HCM heart diseases. In: *2018 fifth HCT information technology trends (ITT)*. IEEE

Implementation of an Automatic EEG Feature Extraction with Gated Recurrent Neural Network for Emotion Recognition



Rajeswari Rajesh Immanuel and S. K. B. Sangeetha

Abstract Emotion is a complicated state that influences one's thoughts and behaviour. Recognizing the emotions of a human being is a major research interest in the affective computing after this pandemic situation and which can be applied in medical related fields to cure physical and mental illness. Early detection of stress helps humans to avoid or prevent many diseases related to it. The development of an emotion recognition system using machine learning algorithms has taken a lot of time and effort for researchers and is less focused with Electroencephalography (EEG) signals because EEG signals are noisy, non-linear, and non-stationary. Deep learning algorithms are the most popular solution due to its ability to see images as data. In this paper, we propose a deep learning framework, Gated Recurrent Unit Emotion Recognizer (GRUER) that can detect human emotion with help of EEG signals. This is achieved by implementing four feature extraction algorithms such as Short-Time Fourier Transform (STFT), Wavelet Entropy, Hjorth and Statistical features on dataset and the feature selection method Principal Component Analysis (PCA) is applied to the extracted features to select most significant features to obtain high-accuracy emotion recognizing model. Keras libraries are used to train the model in an appropriate way so that it is neither overfit nor underfit with the data using the Early-stopping function. The performance of the GRUER model is measured using performance metrics such as accuracy, precision, recall and F1-Score are illustrated in the results. The accuracy of the GRUER is 98% and it is a 3-dimensional model which has valence, arousal and dominance for emotion detection. The model loss obtained by GRUER is 1.12 which is low when compared to other models. Finally, the suggested method and its results show that this proposed method outperforms numerous existing emotion recognition systems.

R. R. Immanuel (✉) · S. K. B. Sangeetha
Department of Computer Science and Engineering, SRM Institute of Science and Technology,
Vadapalani, Chennai, India
e-mail: rr6890@srmist.edu.in

S. K. B. Sangeetha
e-mail: sangeets8@srmist.edu.in

Keywords Emotion recognition · Deep learning · EEG signals · Classification · CNN · Stress · Dataset · Feature extraction

1 Introduction

In broader applications, such as mood monitoring, brain state detection is used so that a major human–computer interface platform can change its presentation to the user’s current state of mind. Machine learning faces various challenges when dealing with brain imaging data. Brain signals are noisy, i.e. A multitude of things contaminate the data, including muscle activity, measurement inaccuracy, non-interesting brain activity, and electromagnetic interference from the surroundings. Hundreds of thousands of features can be found in the raw data from each trial. Because the data is a time series with spatial interactions, it may be necessary to investigate temporal components in addition to spatial analysis [1]. Furthermore, data collection takes time and effort. As a result, gathering enough data to run many of the most effective machine learning algorithms can be difficult. The activity of the brain varies greatly between people and even between data collection sessions within a single person. As a result, classifiers can cope with large. However, in brain imaging activities, producing more than a few hundred trials per person is sometimes unfeasible and quantities of variation are necessary. To solve these challenges, researchers used the effectiveness of several deep learning architectures in detecting brain signals [2, 3].

The capacity to train a network with numerous layers was critical for deep learning neural networks to continue to grow. To train a multiple layered neural network, the gradient loss function is used to adjust the weights and the backpropagation also helps in periodical training for the neural network. However, because of a lack of raw processing power and ways to deal with the vanishing gradient problem, which occurs when the back propagated error gradient approaches zero leads to poor performance of neural networks. The neuron in every layer is linked to next in Conventional neural network research which focused on fully linked layers. Earlier these types of layers were used but now due to their training problems, they are not effectively used [4].

The EEG signal is multidimensional because it has both spatial and temporal covariance. Traditional machine learning techniques struggle to account for the high dimensionality, creating a desire to identify signal properties to minimise dimensionality. The non-stationary nature of the EEG signal, however, causes problems for many time series techniques for feature extraction. Deep learning has mainly been investigated in the same way, with each study focusing on a single sort of architecture and a single dataset. There is a scarcity of knowledge on how alternative architectures compare. Furthermore, using EEG data to train a deep learning classifier is a tough task [5, 6]. Feature extraction is currently carried out using time-consuming and labor-intensive theory-driven manual analytic approaches. The suggested study would look into a new deep learning categorization model that uses automated feature extraction from EEG data. Several machine learning methods have been used to automatically extract features from EEG, according to the literature review, but no proper

comparison research of traditional and automatic feature extraction methodologies has been described. The features retrieved from the EEG data are sent to Gated Recurrent Unit Emotion Recognizer (GRUER). The proposed work is evaluated and compared with existing techniques for comparative analysis.

The remainder of the paper is organized as follows: Sect. 2 describes the background of the research domain as well as other related papers. In Sect. 3, the experimental setup, analysis, feature extraction, and classification approaches are all described in depth. The results as well as the discussion are explained in Sect. 4. The conclusion part contains a discussion of the limitations and future works.

2 Related Study

In contrast to machine learning, which processes a large volume of data in a linear manner, the deep learning approach concentrates on the problem in a non-linear manner. As a result, the deep learning method was chosen to investigate emotions, particularly stress. Many studies on Human–Computer Interface (HCI) have been undertaken in recent years, with an emphasis on emotion recognition utilising EEG signals. Because EEG signals are chaotic, non-linear, and non-stationary, research on emotion identification focuses on brain signals rather than EEG signals [7, 8]. Emotion recognition is carried out with general lifestyle features like gender, age, habits conducted with KNHANES VI dataset wherein Stress is classified using Deep Belief Network (DBN). The DBN algorithm is compared with other conventional machine algorithms such as Support Vector Machine (SVM), Naive Bayes Classification (NBC) and Random Forest (RF). The specificity and accuracy in predicting the stress among people are 75.32% and 66.23% respectively. The first layer of the model is dedicated to data collecting and data cleaning, while the second layer is dedicated to data analysis. Feature extraction is performed using t-test and Chi-square test. Finally, the extracted features are fed to DBN and the model predicts the stress [9].

Speech is a way to communicate the thoughts of humans. Hence, speech signals were used to detect emotions of the 56 subjects with the help of RNN and algorithm to obtain the maximum accuracy with Long Short Term Memory (LSTM). Using preprocessed speech samples, the system first obtains mel-filterbank coefficients. Then the output is derived using binary decision criterion using LSTM and the accuracy of the model was 66.4% [10]. Work related stress among selected Smart factory workers was conducted with DAIC-WOZ dataset with 116 audio recordings. Smart factories are digital environments that are highly automated with a cyber ecosystem. Speaker diarization and sampling strategy techniques are used for noise reduction in the audio samples. The Convolutional Neural Network (CNN) and Growing Self-Organizing Map (GSOP) algorithms are used to obtain the prediction of stress with 82% accuracy [11].

Social media posts are also taken as a dataset to identify whether the person is emotionally balanced. Depression lexicon is manipulated and used the same to derive the conclusion. Kaggle.com dataset from Twitter is used to find post depression of

cancer survivors by their Twitter posts. CNN is compared to other algorithms such as MLP, NBC, SVM, CNN n-gram. The accuracy achieved by this model is 91.29%. In this paper, they have used a text analysis tool to find the depression lexicon in the tweet. Then the extracted data is fed to filtering and to CNN layers. The output is whether the cancer survivor is stressed or not [12]. Deep learning approaches are applied in criminals' psychological problems by recording their EEG signals. Around 109 criminal EEG signals are recorded. The EEG is preprocessed using a time-domain regression approach to denoise it and extracted EEG signal features are given to the Back Propagation (BP) algorithm and obtained emotion recognition is 93.5%. The recorded dataset helps to evaluate the stress level in them with MATLAB Tool. The BP algorithm performance is compared with the Bayes classifier, SVM and Hidden Markov Model (HMM) [13].

The study proposes using a late-fusion strategy with two phases to merge acoustic and textual information. The acoustic and text features from the speech have been fed to the SVM to find the Arousal-Valence-Dominance accuracy of the model. The dataset used in this model are from IEMOCAP and MSP-IMPROV datasets and measuring devices used here are Biosemi Active and The Emotiv EPOC. The accuracy of emotion recognition is compared between two datasets. It also compares SVM to Recurrent Neural Network (RNN)—LSTM [14]. Table 1 shows the comparison of recent works done in EEG-based deep learning algorithms. Most of the current applications have feature extraction which is followed by a classifier. Less time implementation and less problem in overfitting in these algorithms than deep learning. SVMs (Support Vector Machines) are an example of this type of model. Another extensively used classifier is Linear Discriminant Analysis (LDA), which produces linear decision surfaces using probabilistic modelling and the requirements of normality and homoscedasticity (same variance are used by all the classes). In order to determine the label, K Nearest Neighbors examines the training data using a distance metric to identify the K samples nearest to the one being categorised. Naive Bayes is a simple probabilistic classifier that makes choices based on maximum likelihood [15].

Fourier Transformations (FT), Wavelet Transforms (WT), PCA, Independent Component Analysis (ICA), autoregressive methods, and combinations of those techniques are the most frequent feature extraction methods. The Fourier transform is likely the most widely used approach for obtaining features. At the expense of temporal data, it gives more information on the data's frequency spectrum. In practise, short-time Fourier transforms are more commonly used to combat this. The Fourier transform is used to determine power spectra, which correlate to EEG literature on brain activity that occurs in separate frequency bands. The wavelet transform is another prominent signal processing method for feature extraction. Wavelet transform techniques also break the signal down into frequency-based information, but unlike the Fourier transform, the model takes into account time and frequency trade-offs [20]. EEG signal feature extraction techniques are less used in deep learning research.

Table 1 Related work comparison

S. No.	Author name	Input	Output	Method/algorithm	Performance metric/accuracy	Content	Dataset
1	Asgar et al. [16]	EEG Signals 2D features Spatial Temporal	Emotion detection	DNN	Accuracy SEED-93.8% DEAP-77.4%	K-mean cluster algorithm is used	DEAP SEED dataset
2	Ming et al. [17]	EEG signals	Emotion recognition	CNN	Accuracy compared with various parameters	Reinforcement learning	37 subjects manually data collected
3	Xing et al. [18]	EEG signals	Emotion recognition	LSTM over SVM SAE over ICA	Accuracy valence-81.10% Arousal-74.38%	Optimizer algorithm is used for reduced gradient descent	Deap dataset
4	Yi Ding et al. [19]	EEG signal Feature Temporal Spatial	Emotion detection	SVM	Accuracy-86.03%	SVM is compared with EEGNet, LSTM	18 subjects EEG signals recorded from MUSE EEG headband

3 Methodology

The proposed methodology has various steps involved from data preprocessing to results. The proposed model GRUER uses EEG signal and involves numerous techniques which are shown in Fig. 1. The dataset is preprocessed by removing artifacts and passed to the feature extraction section. In feature extraction, different required features are extracted. The dimensionality reduction is also done with the PCA and the final vector of data is given to the GRU classifier to classify the different emotions [21].

3.1 Preprocessing

EEG signals contain a lot of noise such as Electrooculogram (EOG) blinking of eyes and Electromyogram (EMG) movement of muscles. The 50–60 Hz power line frequencies act as noise to EEG signals. Below 40 Hz, frequencies relevant for identifying emotions can be found. The preprocessing EEG signals remove artifacts and select the dominant frequency [22]. The EEG dataset was principally preprocessed by filtering with a cutoff frequency of 70 Hz and down sampling to 200 Hz from a sampling rate of 1000 Hz. Figure 1 depicts the effect of the preprocessing stages on the raw EEG signal [23]. The preprocessed EEG signals are passed to the feature extraction module where the significant features related to emotion recognition are extracted.

3.2 Feature Extraction

The EEG-based emotion detection model has a feature extraction stage which identifies the most important characteristics of the dataset available which will help to map

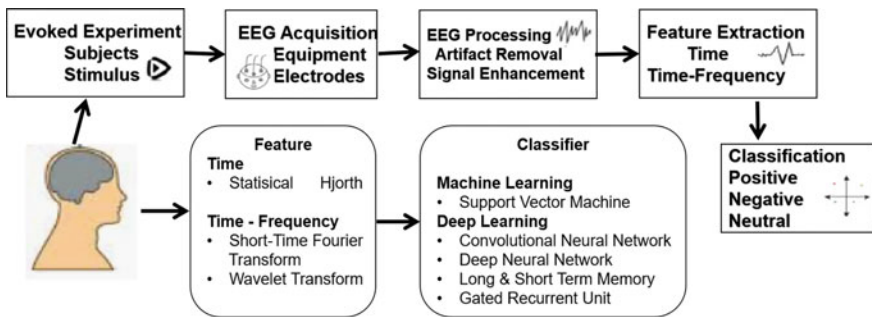


Fig. 1 Proposed model architecture

Table 2 EEG signals band

Sr. no.	Frequency of EEG signals in (Hz)	Band of EEG
1	1–4	δ
2	4–7	Θ
3	8–15	α
4	15.5–16	β ₁
5	16.5–20	β ₂
6	20.5–28	β ₃
7	25–100	γ
8	Potentially evoked	λ
9	Potentially evoked	P300

the emotion properly and it is the key objective for the proposed system [24]. The EEG signal is a random, non-stationary signal that is challenging to extract only in the temporal domain. We employed time–frequency analysis to capture the dynamic change of EEG in the time and frequency domain. We extract the power spectrum feature, Hjorth, and wavelet energy entropy feature for positive, neutral, and negative emotions [25].

3.2.1 Short-Time Fourier Transform (STFT)

One of the ways for revealing the frequency contents of EEG signals at each time point is STFT analysis. This information from the EEG signals extracted using STFT will be helpful in controlling and performing various tasks using HCI. Hence, the STFT is used to extract the features and the different frequencies of EEG signals and their bands are given in Table 2.

STFT is defined as follows:

$$\sum_{k=-\infty}^{\infty} x(k)\omega(k - nR)e^{-j\omega m k} \tag{1}$$

where $x(k)$ is the input signal of the time n , $\omega(n)$ is a window function whose length is m , j is the Bias value [24].

3.2.2 Wavelet Entropy Feature

Entropy is one of the powerful algorithms developed for the non-linear analysis of EEG signals as non-linear dynamics has progressed [24]. The discrete wavelet transform may adapt the window size automatically based on the frequency. The discrete wavelet transform is defined mathematically as follows:

Table 3 The Hjorth feature

Parameter of Hjorth feature	Notation
Activity of parameter	$\text{var}(x(i))$
Mobility of parameter	$\sqrt{\frac{\text{var}(x'(i))}{\text{var}(x(i))}}$
Complexity of parameter	$\frac{\text{mobility}(x'(i))}{\text{mobility}(x(i))}$

$$Wx(i, j) = \sum_j x(t) \underline{\Psi_{i,j}(t)} \quad (2)$$

$Wx(j, k)$ is the data after convolution with scale function $i, j(t)$: where $x(t)$ is EEG data, i is the number of wavelet decomposition layers, and $Wx(i, j)$ is the data after convolution with scale function $i, j(t)$:

$$\Psi_{i,j}(t) = \frac{1}{\sqrt{2^j}} \left(\frac{t}{2} - j \right) \quad (3)$$

3.2.3 Hjorth Parameter

The Hjorth parameter is one way of expressing statistical features of a signal in the time-domain, and it has three types of parameters shown in Table 3. Additionally, they can be used to produce lower computing complexity [26].

3.2.4 Statistical Features

To recognize the emotion, the time series of EEG signals which has a set of signal statistics data are used. From EEG signals, five statistical features are extracted for each channel producing 70-dimensional (14(channels) \times 5(statistics) = 70). This statistical feature vector is used to characterize the EEG signals.

Mean from Statistical Feature

$$y = \frac{1}{M} \sum_M^{m=1} Y(m) \quad (4)$$

Standard Deviation from Statistical Feature

$$\sigma_y = \sqrt{\frac{1}{M} \sum_M^{m=1} Y(m - \mu_y)^2} \quad (5)$$

Mean of absolute values of first difference

$$\theta_y = \frac{1}{M-1} \sum_M^{m=1} |Y(m+1) - Y(m)| \quad (6)$$

Mean of absolute values of second difference

$$\gamma_y = \frac{1}{M-2} \sum_{M-2}^{m=1} |Y(m+2) - Y(m)| \quad (7)$$

Mean of absolute values of second difference of normalized)

$$\gamma_y - \frac{1}{M-2} \sum_{M-2}^{m=1} |Y(m+2) - Y(m)| \quad (8)$$

These parameters can be extracted from the statistics features and it can be used to recognize the emotions of human beings more accurately.

3.2.5 Feature Selection Method—PCA

PCA is a multivariate analytical method based on linear transformation that is frequently used to minimise data dimensionality, extract valuable information from massive data, analyse variable structures, and so on. In this research, the PCA approach was utilised to reduce the dimensionality of the EEG signals. Because the EEG signal's spatial resolution is inadequate, using all channels for feature extraction only adds to the workload. We can use PCA in this circumstance since it can determine the largest variation from the higher input dimensions. As a result, a set of EEG signals can be turned into a single signal using PCA to reduce the number of channels of interest. Using PCA, the selected EEG channels are AF4, F4, F8, FC6, T8, P8, O2, O1, P7, T7, FC5, F7, F3 and AF3. These are the channels of input taken from electrodes that are placed in a particular position on the surface of the head with the help of an EEG recorder. If the data channel has $Y = (Y_1, Y_2, Y_3 \dots Y_n)$ that can be applied to PCA. The data vector Y can be transformed to X by applying PCA to the same.

$$X = U^T Y \quad (9)$$

where U^T is the eigenvector of the covariance matrix of Y . Hence, the elements which are more significant are selected using this process and the selected features which have high information are fed to the classifier.

3.3 Classifier

We have used the Gated Recurrent Unit (GRU) architecture which is similar to the Recurrent Neural Network (RNN) model in deep learning for classifying whether the person is positive (Happy), Neutral (relaxed) and negative (Sad). GRU is a classifier implemented using Keras library. GRU has an input layer, reset gate, update gate and an output layer. GRUER (Gated Recurrent Unit Emotion Recognizer) receives the features extracted from the EEG signal using feature extraction techniques and feature selection techniques. GRUER has an input layer, hidden layer and an output layer which gives the results. We have trained the GRUER model with the dataset which is splitted into training data and test data as 7:3 ratio accordingly.

GRUER consists of different layers and the summary of the model which is the output is shown in Fig. 2.

The input layer consists of the parameters extracted from the process of preprocessing, feature extraction and feature selection of EEG signals. From the input layer, the input is passed to the dense neural network GRUER where the actual process takes place. The neurons in the dense layer or the hidden layer get activated with the help of activation function Softmax. Then, from the activation layer, the feature vector is flattened. The single layer vector is fed to the fully connected layer and the output is the final classification of emotion recognition whether the person is sad, relaxed or happy as shown in Fig. 3.

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 2548, 1)]	0
gru (GRU)	(None, 2548, 256)	198144
flatten (Flatten)	(None, 652288)	0
dense (Dense)	(None, 3)	1956867

Total params: 2,155,011
 Trainable params: 2,155,011
 Non-trainable params: 0

Fig. 2 Different layers involved in GRUER model

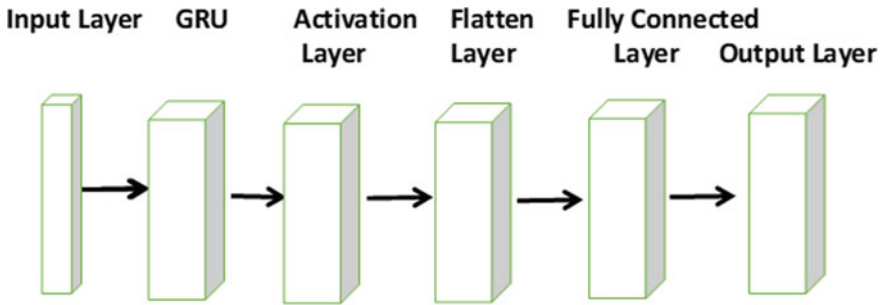


Fig. 3 GRU model architecture of different layers

4 Results and Discussions

4.1 Dataset

In this experiment, the EEG signal is collected by using an EEG Signal recorder and there are 14 channels for collecting EEG across the entire brain. The 12 participants in the age group 25 to 30 participated in the experiment and subjects were shown movie clips with different emotions. The EEG signals are recorded when they were watching the different movie clips for 2 min. The electrodes are placed around the head according to the international 10–20 system. Every electrode can be represented with a letter and a number that identify the lobe and hemisphere in which it should be used (the letters F, T, P, C, and O refer to the frontal, temporal, parietal, central, and occipital lobes, respectively). The selected EEG channels are AF4, F4, F8, FC6, T8, P8, O2, O1, P7, T7, FC5, F7, F3 and AF3 are used for processing. The stimuli is collected from the participants when they watch different movie clips with different emotions. This dataset is taken from the deep database shown in Fig. 4 and the preprocessed EEG signal is plotted using the Matplot function in Fig. 5.

4.2 Implementation

For this experiment, we have used Keras library in Tensorflow to increase the speed of the execution of the proposed GRUER model. The speed of the execution of the deep learning network is faster and easier with Keras. We have also implemented an Early-Stopping function to reduce the number of Epoch and also to monitor the loss in the model as explained in Fig. 6. Overfitting or underfitting can be avoided using the Early-Stopping function. The model is properly trained with the dataset.

The loss function used in this model is categorical-cross entropy which is suitable for multi-class classification. GRUER model is a multi-class classification, since it classifies emotion as positive, negative and neutral. Moreover, Softmax activation

	# mean_0_a	mean_1_a	mean_2_a	mean_3_a	mean_4_a
0	4.620	30.3	-356.0	15.60	26.3
1	28.800	33.1	32.0	25.80	22.8
2	8.900	29.4	-416.0	16.70	23.7
3	14.900	31.6	-143.0	19.80	24.3
4	28.300	31.3	45.2	27.30	24.5
...
2127	32.400	32.2	32.2	30.80	23.4
2128	16.300	31.3	-284.0	14.30	23.9
2129	-0.547	28.3	-259.0	15.80	26.7
2130	16.800	19.9	-288.0	8.34	26.0
2131	27.000	32.0	31.8	25.00	28.9

2132 rows × 2549 columns

Fig. 4 Preprocessed and feature extracted dataset

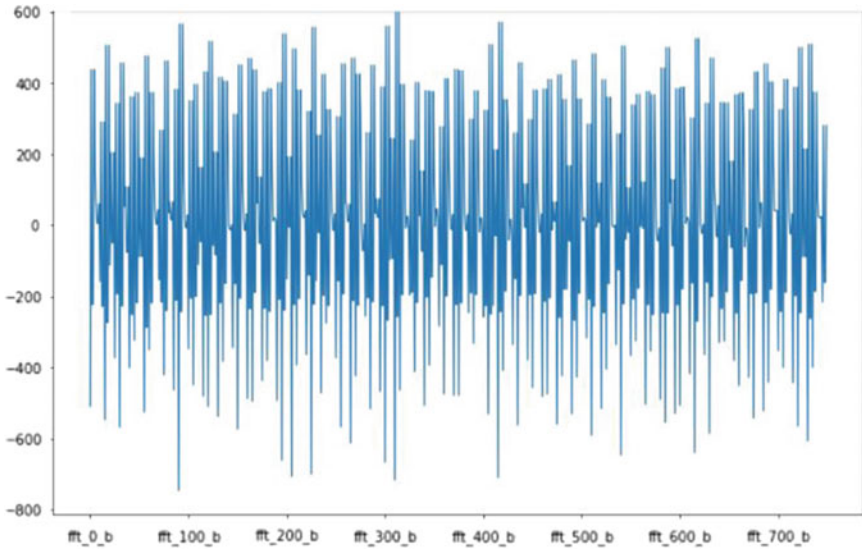


Fig. 5 Preprocessed EEG signals

```
Epoch 00014: val_accuracy did not improve from 0.97812
Epoch 15/40
47/47 [=====] - 552s 12s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 1.1602 - val_accuracy: 0.9781

Epoch 00015: val_accuracy did not improve from 0.97812
Epoch 16/40
47/47 [=====] - 553s 12s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 1.1602 - val_accuracy: 0.9781

Epoch 00016: val_accuracy did not improve from 0.97812
Epoch 17/40
47/47 [=====] - 552s 12s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 1.1602 - val_accuracy: 0.9781

Epoch 00017: val_accuracy did not improve from 0.97812
Epoch 18/40
47/47 [=====] - 551s 12s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 1.1602 - val_accuracy: 0.9781

Epoch 00018: val_accuracy did not improve from 0.97812
Epoch 19/40
47/47 [=====] - 553s 12s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 1.1602 - val_accuracy: 0.9781

Epoch 00019: val_accuracy did not improve from 0.97812
Epoch 20/40
47/47 [=====] - 555s 12s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 1.1602 - val_accuracy: 0.9781

Epoch 00020: val_accuracy did not improve from 0.97812
Epoch 00020: early stopping
```

Fig. 6 Validation accuracy and loss calculation

function is used with categorical- cross entropy as they both work well together [15]. Optimizers are used in deep learning to control the parameters of a neural network so that the cost function is minimised. In this model, Adam optimizer is added to improve the performance of the GRUER. Figure 8 reveals that the model loss is compared between train data and test data. The loss for the training data slowly decreases whereas for test data, it is low from the beginning as already the training is done with the train data. Finally, the training data loss is reduced to zero and for the test data, loss is 1.16 respectively shown in Fig. 6.

The model accuracy is described in Fig. 7. The model accuracy graph is compared with train data and test data respectively. The X-axis consists of a number of epochs and the Y-axis consists of an accuracy percentage upto 1.0. As the number of epoch increases the accuracy also increases. For train data, accuracy achieved is 1.0 whereas for test data, around 10 epochs, it reaches the stagnate accuracy of 0.98. Early-stopping function should in this situation to properly train the system neither overfit nor underfit the model.

Fig. 7 Analysis of model accuracy

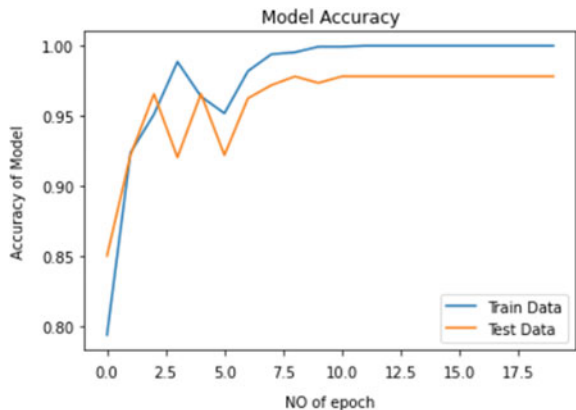
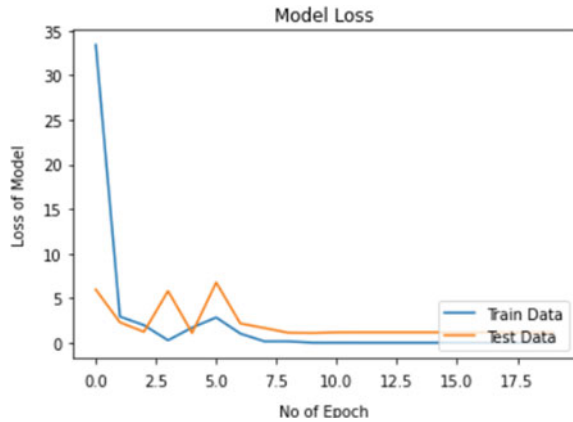


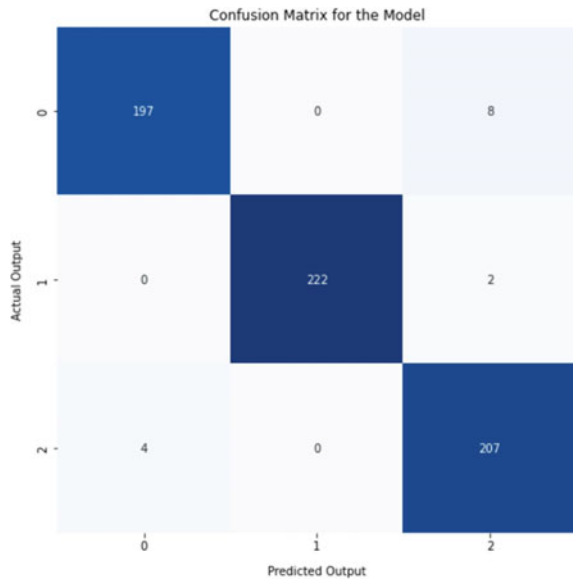
Fig. 8 Analysis of model loss



The Confusion matrix is plotted for the proposed model in Fig. 9 has actual output against predicted output. The labels for emotions positive, negative and neutral are represented as 0,1,2 respectively. The other performance metrics are also plotted in Fig. 10 as precision, recall and F1-score respectively. The final accuracy for the proposed model GRUER is 0.98 which is higher than other previous models.

The deep learning or machine learning algorithms are compared with the proposed model. The performance of the various classifiers such as DNN, CNN, LSTM and SVM is compared with GRUER for accuracy and model loss. Eventually, the proposed method and its outcome presents better accuracy and model loss which is more important is shown in Table 4. The proposed model gives 98.0% accuracy

Fig. 9 Confusion matrix for the accuracy of the model GRUER



Classification Report For Different Performance Metrics:

```

-----
              precision    recall  f1-score   support

     0           0.98       0.96       0.97         205
     1           1.00       0.99       1.00         224
     2           0.95       0.98       0.97         211

 accuracy                   0.98         640
 macro avg           0.98       0.98       0.98         640
 weighted avg        0.98       0.98       0.98         640

```

Fig. 10 Performance metrics for the model GRUER

Table 4 Performance comparison of different models with GRUER

Sr. No.	Classifiers/method	Performance metrics/accuracy (%)	Model loss
1	DNN (Asghar et al. [16])	93.8	1.50
2	CNN (Ming et al. [17])	90.5	1.85
3	LSTM (Xing et al. [18])	81.1	2.2
4	SVM (Yi Ding et al. [19])	86.03	1.80
5	Proposed Model (GRUER)	98.0	1.16

and lowest model loss of 1.12 for validation data. Figure 11 depicts the performance graph of all the classifiers with the accuracies respectively. The proposed model shows higher accuracy than other models in the row. The X-axis displays accuracy and Y-axis displays the different models namely CNN, DNN, LSTM, SVM and GRUER. The proposed model illustrates higher performance by 4% higher than the existing better model for validation loss. Figure 12 represents model loss for the above said models. The GRUER exhibits the lowest model loss among all the five models.

5 Conclusion

In this study, we propose a deep learning system called GRUER (Gated Recurrent Unit Emotion Recognizer), which uses EEG signals to recognize human emotion. This is accomplished by applying four feature extraction algorithms to the dataset, including STFT, Wavelet Entropy, Hjorth, and Statistical features, and using the feature selection method principal component analysis (PCA) to select the most significant features to obtain a high-accuracy emotion recognizing model. To distinguish three types of emotional states, an EEG-based emotion classification approach GRUER based on fused characteristics was applied (neutral, happiness and sadness) and GRU dense neural network is used for the classification. The accuracy obtained

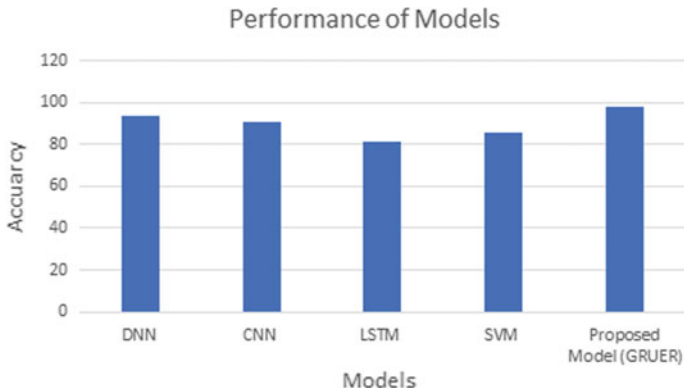


Fig. 11 Performance of various models with GRUER model

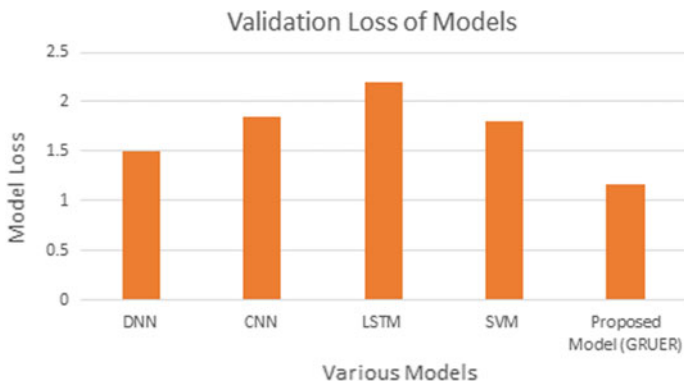


Fig. 12 Validation loss for different models with GRUER model

from this model is 98% and model loss is 1.12 for validation data. Feature selection method used for selecting the important features from the extracted features with the help of PCA which performs better for EEG signal. The same dataset is executed with different deep learning and machine learning algorithms. The performance of different algorithms are compared with the GRUER model and the proposed model provides higher performance than others.

References

1. Rahman A, Hossain F, Hossain M, Ahmed R (2020) Employing PCA and t-statistical approach for feature extraction and classification of emotion from multichannel EEG signal. *Egypt Inf J* 21(1):23–35. ISSN 1110-8665 <https://doi.org/10.1016/j.eij.2019.10.002>
2. Mohammad A, Siddiqui F, Afshar Alam M (2021) Feature extraction from EEG signals: a deep

- learning perspective. In: 11th international conference on cloud computing, data science & engineering (Confluence), pp 757–760. <https://doi.org/10.1109/Confluence51648.2021.9377108>
3. Gao Q, Wang C-H, Wang Z, Song X-L, Dong E-Z, Song Y (2020) EEG based emotion recognition using fusion feature extraction method. *Multimed Tools Appl* 79:27057–27074 (2020). <https://doi.org/10.1007/s11042-020-09354-y>
 4. Islam MR, Barua S, Ahmed MU, Begum S, Di Flumeri G (2019) Deep learning for automatic EEG feature extraction: an application in drivers' mental workload classification. In: Longo L, Leva M (eds) *Human mental workload: models and applications*. H-WORKLOAD 2019. *Communications in computer and information science*, vol 1107. Springer, Cham. https://doi.org/10.1007/978-3-030-32423-0_8
 5. Khare SK, Bajaj V (2021) Time–frequency representation and convolutional neural network-based emotion recognition. *IEEE Trans Neural Netw Learn Syst* 32(7):2901–2909 (2021). <https://doi.org/10.1109/TNNLS.2020.3008938>
 6. Nawaz R, Cheah KH, Nisar H, Yap VV (2020) Comparison of different feature extraction methods for EEG-based emotion recognition. *Biocybern Biomed Eng* 40(3):910–926. ISSN 0208-5216. <https://doi.org/10.1016/j.bbe.2020.04.005>
 7. Shon D, Im K, Park J-H, Lim D-S, Jang B-T, Kim J-M (2018) A novel EEG feature extraction method using Hjorth parameter. *Int J Electron Electr Eng* 15(11):2461. <https://doi.org/10.3390/ijerph15112461>
 8. Sangeetha SKB, Dhaya R, Shah DT, Dharanidharan R, Praneeth Sai Reddy K (2021) An empirical analysis of machine learning frameworks digital pathology in medical science. In: *Journal of Physics: Conference Series*, vol 1767, p 012031. <https://doi.org/10.1088/1742-6596/1767/1/012031>
 9. Song S-H, Kim D (2017) Development of a stress classification model using deep belief networks for stress monitoring. *Healthcare Informatics Research*. 23:285. <https://doi.org/10.4258/hir.2017.23.4.285>
 10. Han H, Byun K, Kang H-G (2018) A deep learning-based stress detection algorithm with speech signal. In: *Proceedings of the 2018 workshop on audio-visual scene understanding for immersive multimedia (AVSU'18)*. Association for Computing Machinery, New York, NY, USA, pp 11–15. <https://doi.org/10.1145/3264869.3264875>
 11. Madhavi I, Chamishka S, Nawaratne R, Nanayakkara V, Alahakoon D, De Silva D (2020) A deep learning approach for work related stress detection from audio streams in cyber physical environments. In: 25th IEEE international conference on emerging technologies and factory automation (ETFA), pp 929–936. <https://doi.org/10.1109/ETFA46521.2020.9212098>
 12. Ismail NH, Liu N, Du M, He Z, Hu X (2020) A deep learning approach for identifying cancer survivors living with post-traumatic stress disorder on Twitter. *BMC Med Inf Decis Mak* 20. <https://doi.org/10.1186/s12911-020-01272-1>
 13. Liu Q, Liu H (2021) Criminal psychological emotion recognition based on deep learning and EEG signals. *Neural Comput Appl* 33:433–447. <https://doi.org/10.1007/s00521-020-05024-0>
 14. Atmaja BT, Akagi M (2021) Two-stage dimensional emotion recognition by fusing predictions of acoustic and text networks using SVM. *Speech Commun* 126:9–21. ISSN 0167-6393. <https://doi.org/10.1016/j.specom.2020.11.003>
 15. Abdulkarim H, Al-Faiz MZ (2021) Online multiclass EEG feature extraction and recognition using modified convolutional neural network method. *Int J Electr Comput Eng* 11(5). <https://doi.org/10.11591/ijece.v11i5.pp4016-4026>
 16. Asghar M, Khan M, Rizwan M, Mehmood RM, Kim S-H (2020) An innovative multi-model neural network approach for feature selection in emotion recognition using deep feature clustering. *Sensors* 20. <https://doi.org/10.3390/s20133765>
 17. Ming Y, Wu D, Wang Y, Shi Y, Lin C (2020) EEG-based drowsiness estimation for driving safety using deep q-learning. *IEEE Trans Emerg Top Comput Intell*. <https://doi.org/10.1109/TETCI.2020.2997031>
 18. Xing X, Li Z, Xu T, Shu L, Hu B, Xu X (2019) SAE+LSTM: a new framework for emotion recognition from multi-channel EEG. *Front Neurorobot* 13. <https://doi.org/10.3389/fnbot.2019.00037>

19. Ding Y, Robinson N, Zeng Q, Chen D, Wai AAP, Lee TS, Guan C (2020, July) Tsception: a deep learning framework for emotion detection using EEG. In: 2020 international joint conference on neural networks (IJCNN) (pp. 1–7). IEEE.
20. Roy Y, Banville H, Albuquerque I, Gramfort A, Falk TH, Faubert J (2019) Deep learning-based electroencephalography analysis: a systematic review. *J Neural Eng* 16(5). <https://doi.org/10.1088/1741-2552/ab260c>
21. Siddiqui MK, Morales-Menendez R, Huang X, Hussain N (2020) A review of epileptic seizure detection using machine learning classifiers. *Brain Inf* 7(1):5. <https://doi.org/10.1186/s40708-020-00105-1>
22. <https://keras.io/about/>
23. <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>
24. <https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/>
25. https://keras.io/api/callbacks/early_stopping/
26. <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy>

High Performance Classifier for Brain Tumor Detection Using Capsule Neural Network



J. S. Thanga Purni, R. Vedhapriyavadhana , S. L. Jayalakshmi ,
and R. Girija 

Abstract Brain tumor is a kind of dangerous disease that continues to spread worldwide. Therefore, early diagnosis and treatments are extremely important in this case, and it is considered the most common form of cancer that can be found in children and adults. Finding out the right type of tumor in the early stages is an important part for developing a specific treatment and diagnosing the patient's response according to the treatment. In this paper, we propose a Caps Net (Capsule Neural Network) model to avoid the rise of mortality among the brain tumor people and to reduce the time required for accurate diagnosis. The proposed method involves MRI images for classifying the different types of tumors and outperforms CNN in accuracy.

Keywords Brain tumor · Accuracy · Capsule neural network · Early diagnosis malignant and benign

1 Introduction

In this era, securing information is a challenging task. Information also includes audio [1, 2], video [3, 4] and image files [5–9]. A proper cryptosystem must be designed to protect the information. It's not only important to design the cryptosystem, whereas the designed cryptosystem must be able to withstand attacks. Many researchers [10]

J. S. T. Purni · R. Vedhapriyavadhana (✉)
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
e-mail: vedhapriyavadhana.r@vit.ac.in

J. S. T. Purni
e-mail: purni1520@gmail.com

S. L. Jayalakshmi
Department of Computer Science (Main Campus), School of Engineering and Technology,
Pondicherry University, Puducherry, India
e-mail: sathishjayalakshmi02@pondiuni.ac.in

R. Girija
CST Department, Manav Rachna University, Faridabad, India
e-mail: girija.srikanth09@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
R. J. Kannan et al. (eds.), *Computer Vision and Machine Intelligence Paradigms for SDGs*, Lecture Notes in Electrical Engineering 967,
https://doi.org/10.1007/978-981-19-7169-3_14

focused on many technologies in this field. IOT also came into this field for monitoring [11, 12]. When an attacker attacks the system, recognizing the attacker is a challenging task. In such a case, face recognition [13] plays an important role. Nowadays, deep learning and machine learning plays an important role in securing the information [14]. Deep learning concentrates more on health monitoring and diet monitoring [15]. Monitoring health is very much needed for all human beings. In case if any human fails to monitor his/her health, it causes very dangerous diseases will come into the picture. One of the diseases is Brain tumors.

Brain is the utmost complicated organ [16] that controls the mechanism of the human body if any damage occurs that can lead to death because the brain works with many cells. Nowadays, there are many methods available for analyzing the brain tumors and their behavior.

The serious form of brain tumor with hyper-growth of cells is called malignancy. All the brain tumors are not exactly malignant (cancerous), but some tumors are benign (non-cancerous). Primary and secondary cancers are two important categories of brain tumors. Primary brain cancer is a rare form of cancer that starts from the brain itself. This kind of cancer damages [17] the healthy tissues in the brain and spinal cord. Subsequently, it spreads across the body to attack the immune system of the human body. Secondary brain cancer is the most common type of cancer, which starts from any part of the body rather than the brain. For instance, lung cancer or breast cancer spreads to the brain and it is called metastatic brain cancer.

This paper mainly focuses on two things which is to overcome the drawbacks of CNN in the pooling layer which is mainly needed for segmentation and detection and the time required for the detection of tumor from the input image by overcoming these two major problems it will be much help on the doctor's side to plan the required treatments for their patient and analyzes their response. Most of the conventional methods such as Support Vector Machine (SVM) and Artificial Neural Networks (ANN), Naive Bayes (NB) [18–20] were able to detect tumor in brain with very lesser percentage in the early stage and it leads to a person's death.

In this case, if the current treatment plan was not giving much response, they can go for a new treatment plan if the tumor stage has been caught earlier and to avoid unnecessary critical situations for doctors and patients. This model has been proposed to handle such an abnormal situation by detecting the tumor in an early stage to avoid the severity of the disease in human beings.

CNNs have been constructed to classify images with layers of convolutions and pooling. While performing this technique we have a few shortcomings. One among them is that while performing [21, 22] pooling leads to a loss of preferred spatial transformation details and different positional parameters of the inputs. This information plays a vital role in identifying the spot of the tumor while performing object segmentation and detection. Even when we detect the tumor type if we know the location and size of the tumor information only can help doctors to do the operation.

Sometimes while operating the tumor, it may be bigger than the MRI also so as much possible information doctors get about the tumor from input image will be helpful while the patient goes for operation and the second problem is the time required for the detection in classification, either we get more accuracy or time

accurate can be done to solve these two problems at the same time this proposed model can be used. Brain cancer can be treated and cured if caught early which needs higher accuracy while predicting the type of cancer. For examining and testing the spatial values from input pixel and for extracting the features by statistical method [23] GLCM has opted a better performance.

The rest of the paper is organized as follows: Sect. 2 presents a review of methods for brain tumor classification. The proposed approach is presented in Sect. 3. Sections 4 and 5 present the studies carried out and the performance analysis of proposed method.

2 Methods

2.1 Convolutional Neural Network

CNN is the main key for deep learning success. Till today it has been used for computer vision tasks. It is used to identify the different objects from given input image. The main feature of CNN is it requires less pre-processing tasks compared to the other classification algorithms for object detection. CNN is basically the combination of two successive layers called pooling [24, 25] and convolution to fetch the embedded features of an image. The extracted features are used for both training of the model and for classification of images.

2.2 Capsule Networks

Capsule Networks, the process is about routing by agreement which is mainly used for solving the flaw of convolutional neural network in pooling layers. In Capsule Network these layers are replaced by “routing the agreement”, after this process the parent capsule will get the result from the next layer when their coupling coefficients are not getting a similar performance. From the parent capsule each capsule attempt to predict the output, and therefore the coupling coefficient between the two capsule increases when the prediction is consistent with the actual output of the parent capsule. This Model has 64 feature maps within the convolution layer instead of 256.

3 Literature Survey

Heba Mohsen et al. [26], the techniques which have been used here are feature extraction, DNN classifier for tumor classification, image segmentation and reduction using Principle component analysis. Pamian Afshar et al. [27], with the real set of MRI images Caps Net is Investigated for the over-fitting problem and checking their capability for providing the better fit for the whole image or just the segmented part. Sanjeev Kumar et al. [28], measured the training performance, classification accuracies and computational time by using the modified Probabilistic Neural Network Classifier. Which gives rapid and accurate classification and decreases the processing time. Dina Aboul Dahab et al. [29], by using the hybrid approach the manual labelling time and error can be reduced and the features are reduced by applying the Principle Component Analysis technique.

Mesut Togacar et al. [30], built on an architecture with attention modules and hyper column. The BrainMRNet model used in the convolutional layer is hyper column and the features extracted from each layer of BrainMRNet model are retained by the array structure.

The main goal of the proposed model is to focus on the diseased area. Minakshi Sharma et al. [31], ANFIS is an adaptive network which mixes the specification and properties of both fuzzy and neural networks, and a Comparative analysis has been performed, Pankaj Sapra et al. [32], for the training performance, classification accuracies and computational time PNN classifier is used. Mehdi Fatan Serj et al. [33], with the help of Deep-CNN architecture the features of the CT-scan images can be learned and compacted at the preliminary layers of a deep model.

Hojjat Salehinejad et al. [34], with the original imbalanced dataset Augmenting with GAN generated images improve the performance of classification. Zhenghao Shi et al. [35], the approaches are to form a map for ANN techniques for the medical image processing. The result has been shown with a combination of real and artificial data used to train the DCNN.

4 Proposed Model

In the proposed model, the capsule network architecture is implemented by using convolutional layer with reshaping and squashing functions. The tumor types that we considered are Glioma, Meningioma and Pituitary. Glioma occurs in the brain and spinal cord, and it can affect severely the brain function. The criticalness can be found based on the location and its growth rate. Meningioma arises from the meninges, it's the membranes that are surrounded by our brain. Usually it grows very slowly, without any major symptoms. Pituitary tumors are abnormal growths that arise in pituitary gland. Most of these types are benign and these growths remain within the pituitary gland or in the surrounding tissues.

$$\hat{u}_{j|i} = W_{ij}u_i$$

Here u_i will be the output of the child capsule layer has been multiplied with the weighting matrix and from that the prediction output value of parent capsule will be found as $\hat{u}_{j|i}$

Coupling coefficient for calculating the SoftMax function

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

Routing by agreement process

$$s_j = \sum_i C_{ij} \hat{u}_{j|i}$$

And as finally the squash function secures the output vector to form the final output for each capsule in its initial defined value

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}$$

For updating the log prob and coupling Coeff when the two vectors agree after the routing process

$$a_{ij} = v_j \hat{u}_{j|i}$$

4.1 Capsule Network Model Creation

1. Primary capsules

Convolution + Reshape + Squash

The input MRI images are fed into the convolution layers after which the outputs are mapped with the features extracted and stored in an array. It is then further processed by applying the function of reshaping to retain the data even after the new shape is achieved. The vital information is preserved even if the shape is changed. Now the Sigmoidal shaped function which acts as a squashing function is smeared to make sure that the complete range of values lies within the small range variety of having upper limit value of 1 and lower limit value of 0 to ensure that the conditional probability distribution of the ideal frame arrived. And, by maintaining the simplest form of derivatives by just taking the mere difference between the basic functions.

The MRI images archived by the Picture Archiving Communication Systems are stored without deviation of the stational information in the advanced dimension.

2. Higher layer capsules

Routing by agreement

Before the higher layer capsule starts its operation, the capsule lies in the primary layer and will forecast the outcome of the higher layer capsules. For an understanding, it is considered that the number of capsules in the primary layer is 48. Then in such case if 24 is for certain silhouettes and another 24 is for next set of characters.

In the higher layer there are only two types of outcomes expected and now prediction of the higher layer is done by each set of characters. When we combine both predictions and project the predictions which are similar and at the same time proposed by majority of the silhouettes. Now the higher layer as a double check, will also perform the operation and find its own output and cross-check with primary layer. If both the primary and higher layer accepts to pick out the same, then the output capsule is located through similarities with higher prediction accuracy.

Initially the weights are assigned as zero while finding the similarities between the capsule layers. To normalize the same, a normalized exponential function is applied as SoftMax Function to get the exponential terms of the applied input image. The predicted output is consigned in the higher layer and finally calculates the weighted sum of all capsules cumulatively so that changes in weight are done repetitively for every pass till the error gets reduced which is calculated between the predicted and the actual output.

If the predicted do not correlate with the features in the actual output, then the weights added would be very less and do the process once again. Wherever the predictions are sturdy then more weights are involved in the process. The finalization of capsule in the higher layer is fixed after passing through many reiterations. Less number of reiterations leads to less computation time.

3. Decoder

In the Unscrambling decoder, three dense layers are connected, out of which two are piecewise linear functions and the last one is the logistic function for repairing the input image. This network will learn on its own how to reconstruct the input image. Based on the training and adapt itself in identifying the features during testing. It tries to reduce the MSE (Mean Square Error) and do the reconstruction immediately.

4. Loss Calculation

Margin loss—The square loss function is used to find the mismatch between the true and the predicted one.

$$l_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2$$

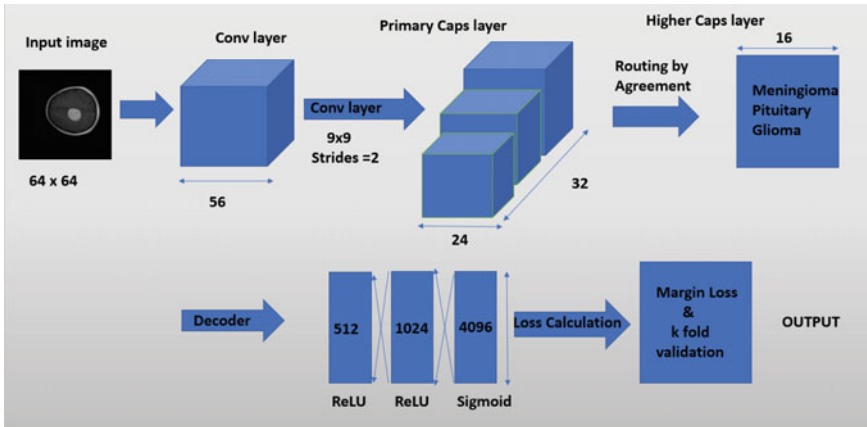


Fig. 1 Caps net architecture

5. Loss due to Refurbishment

The repairing block as well as the calculated rebuilding loss prevents the model to get into the trap of losing information and ensures to keep up the information and stores it safely without encountering it into a loss. The problem of over-fitting is avoided during this phase of training.

The recommended Caps Net Architecture (Fig. 1) is composed of the first layer as the convolutional layer, the resized output after applying the squash function in the primary layer and the consecutive layers of higher layers along with an Un scrambler unit and refurbishment loss function.

In Capsule Network, pooling layers of CNN are exchanged by the method called “direction finding”. The best feature of this Model is it has 64 feature maps within the convolution layer instead of 256. With this created capsule network model through VS code, a dashboard has been created and through that we can insert our input and see the type of tumor in the given image and confidence level of the tumor.

4.2 Prediction

Once the models have been created then through the flask framework with the created html page it will call upon the proposed capsule model and carry on the inputs from users and reshape the image dim into standard level and investigate the model conditions it will predict the tumor type and give the confidence level of the tumor.

– Predicting Confidence level

```
answ = model.predict(x)
classification = np.where (answ == np. amax(answ)) [1][0]
```

```
pred = str(round(answ[0] [classification]*100 ,3)) + '% confidence
there is ' + names(classification)
```

5 Proposed Algorithm

5.1 Prediction Through Flask Framework

– Declare model--> model.h5

```
1. def x(number):
   if (x == 0):
       return classes [0]
   Elif (x == 1):
       return classes [1]
2. user input on X
3. Resize X to 224 x 224
4. model.predict(x)
5. Classification [(0), (1)]
```

6 Results and Discussions

6.1 Accuracy

The obtained accuracy of the Capsule Network Model with the average k-fold validation is 87% with the array of accuracies between 88.44 and 86.87%. K value is 5 since the dataset given is fragmented into 5 groups for validation.

Average K-Fold Validation Accuracy: 0.8766318533968365
 Array of Accuracies: [0.88446475 0.86879896]

6.2 Predicting Tumor

Based on the model a web application is created through which when the data is fed as input through uploading the different stages of tumor is predicted and are discussed as given below.

- **Detecting Pituitary Stage**

Evidence of tumor in the pituitary gland (Fig. 2) which is mostly appeared to be benevolent.

- **Detecting Meningioma Stage**

The meningioma stage is detected with 73.32% confidence (Fig. 3). Meninges grow slowly without any symptoms over a long period of time that surrounds the brain.

- **Detecting No Tumor**

The MRI image with no tumor evidence (Fig. 4) is also proved with 100% confidence level. The probability of the multiple classes is predicted and if the squared length of the object is between 0.1 and 0.9 the prediction cannot be accurate. To

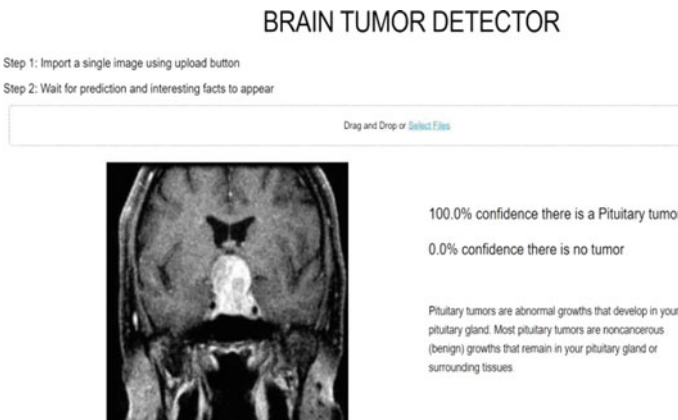


Fig. 2 Pituitary stage

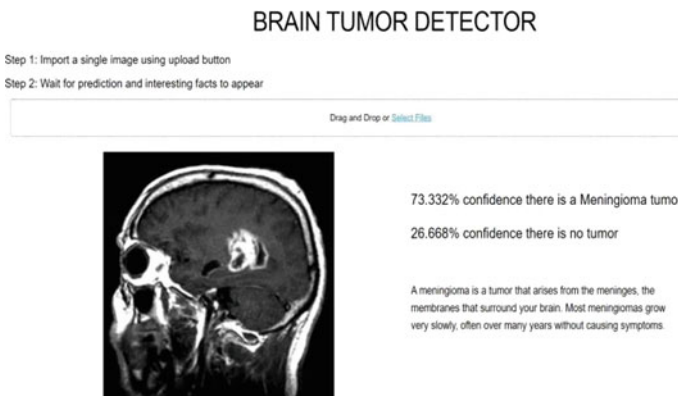


Fig. 3 Meningioma stage

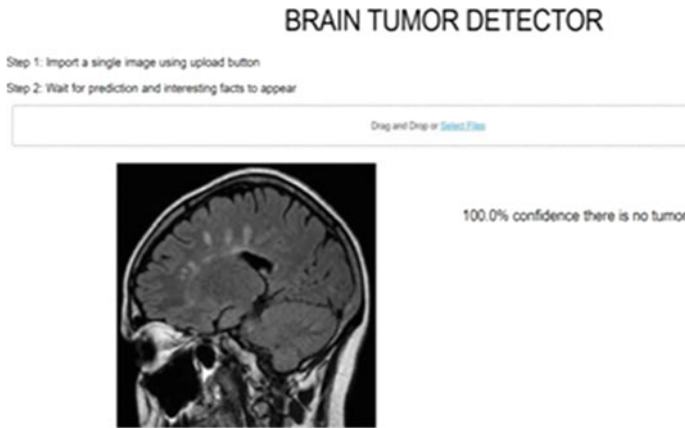


Fig. 4 No tumor

finalize whether the object is present the calculated squared length should not be less than 0.9 and to prove the absence of the object the squared length must not lie below 0.1. So based on the probability calculation of the various classes the loss function is deliberated.

6.3 Training and Validation

The Training accuracy and Validation accuracy for the proposed model (Fig. 5) seem to be linearly increasing. The Training loss and Validation loss for the proposed model (Fig. 6) seem to be linearly decreasing. The Precision, Recall and F1 Score values (Table 1) are calculated for all the three classes.

Fig. 5 Training and validation accuracy

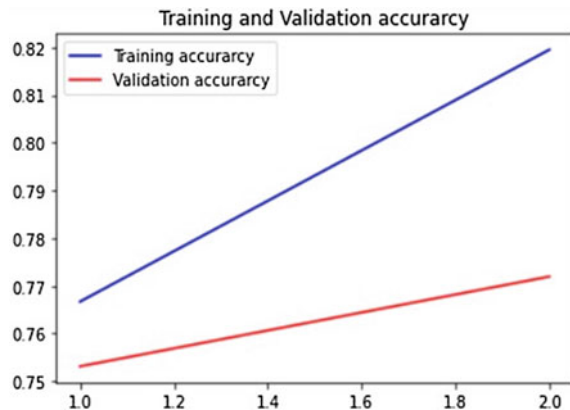


Fig. 6 Training loss versus validation loss

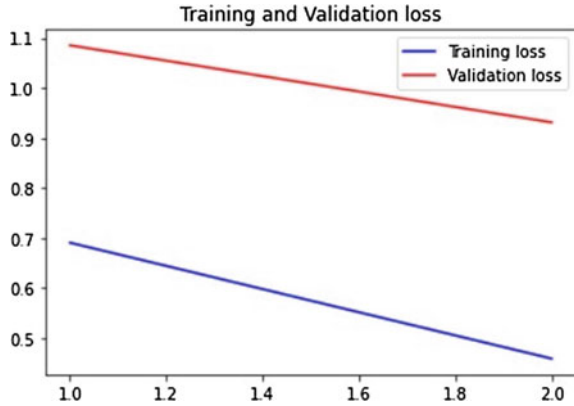


Table 1 Performance metric table

Class	Precision	Recall	F1-Score
Meningioma	87.76	90.44	89.01
Glioma	95.98	87.02	91.28
Pituitary	92.60	97.94	95.19

Table 2 CNN versus caps net accuracy

Class	CNN accuracy	Capsule net accuracy
Meningioma	83.12	91.12
Glioma	85.02	95.02
Pituitary	89.53	96.53

7 Conclusion

In this paper, prediction of Brain tumor has been the greatest challenge for many years, but early diagnosis can be a good way to prevent and diagnose the brain tumor and in image processing model there are various methodologies that are the greatest challenge for many years, but early diagnosis can be a good way to prevent and diagnose the brain tumor and in image processing model there are various methodologies are available and the methods are competing each other to achieve the best output. From that working on the Capsule Neural Network model will lead us to reduce the time required for accurate diagnosis. The accuracy of CNN Model and the proposed Capsule Net module (Table 2) is compared for all the three classes. The CPU computational time is calculated (Table 3) for CNN Model and the proposed Capsule Net module

These capsule networks model can work with minimum number of training samples, and in this network the units are equivariant, it outperforms when compared to CNNs in tumor classification problem.

Table 3 CPU time calculations

Models	CPU computational time (sec)	
	Sample 1	Sample 2
CNN	3.3625	1.7853
Caps net	0.3902	0.1531

```

354/354 [=====] - 182s 514ms/sample - loss: 0.3264 - acc:
0.8701 - val_loss: 0.6070 - val_acc: 0.7105
Epoch 14/15
354/354 [=====] - 162s 459ms/sample - loss: 0.2586 - acc:
0.8644 - val_loss: 0.5835 - val_acc: 0.7632
Epoch 15/15
354/354 [=====] - 147s 416ms/sample - loss: 0.2260 - acc:
0.9209 - val_loss: 0.5967 - val_acc: 0.6711
354/354 [=====] - 50s 142ms/sample - loss: 0.4521 - acc:
0.8418
    
```

Fig. 7 Accuracy-capsule network

The accuracy is predicted as 84.18% in 15 epochs (Fig. 7) using Capsule Network for a batch of 354 datasets.

References

1. Chandrakala S, Jayalakshmi SL (2019) Environmental audio scene and sound event recognition for autonomous surveillance: a survey and comparative studies. *ACM Comput Surv (CSUR)* 52(3):1–34
2. Chandrakala S, Jayalakshmi SL (2019a) Generative model driven representation learning in a hybrid framework for environmental audio scene and sound event recognition. *IEEE Trans Multimedia* 22(1):3–14
3. Nikitha R, Vedhapriyavadhana R, Anubala VP (2019) Video saliency detection using super pixel-based co-segmentation through weight based spatio temporal feature extraction technique. *J Appl Sci Comput* 6(4). April 2019 ISSN No:1076-5131
4. Vedhapriyavadhana R, Shanmuga Priya G, Renee Reddiar N, Mano Priya M (2020) Effective video saliency mapping for object detection using deep learning neural networks. *Int J Recent Technol Eng (IJRTE)* 8(6). March 2020 ISSN: 2277-3878
5. Girija R, Hukum Singh (2020) An asymmetric cryptosystem based on the random weighted singular value decomposition and fractional Hartley domain. *Multimedia Tools Appl* 79(47):34717–34735
6. Girija R, Hukum Singh (2019) Triple-level cryptosystem using deterministic masks and modified Gerchberg-Saxton iterative algorithm in fractional Hartley domain by positioning singular value decomposition. *Optik* 187:238–257
7. Girija R, Hukum Singh (2017) A new substitution-permutation network cipher using Walsh Hadamard Transform. In: 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), IEEE, pp 168–172
8. Girija R, Hukum S (2018) Enhancing security of double random phase encoding based on random S-Box. *3D Res* 9(2):1–20
9. Girija R, Hukum S (2018a) Symmetric cryptosystem based on chaos structured phase masks and equal modulus decomposition using fractional Fourier transform. *3D Res* 9(3):1–20

10. Sangeetha SKB, Jayalakshmi SL (2015) Signcryption approaches for network security. *Int J sci Eng Res (IJSER)* 6(5). May 2015
11. Vedhapriyavadhana RE, Rani SS, Vishnu Kumar VG (2018) Residential and official extension of IOT enabled building automation system. *Indo-Iranian J Sci Res (IJSR) Peer-Rev Quart Int J* 2(2):92–99
12. Priya LR, Vedhapriyavadhana R, Ignisha Rajathi G, Allwyn Kingsley G, Johny Elton R (2021) An IOT integrated fire monitoring and detection system using Raspberry Pi. *Turkish J Physiotherapy Rehabil* 32(3). May 2021 ISSN: 2651-4451
13. Francy Irudaya Rani E, Vedhapriyavadhana R, Jeyabala S, Jothi Monika S, Krishnammal C (2018) Attendance monitoring using face recognition with message alert. *Indo Iranian J Sci Res (IJSR)*
14. Gayathri A, Radhika S, and Jayalakshmi S.L, (2019) Detecting fake accounts in media application using machine learning. Special Issue Published in *Int J Adv Networking Appl (IJANA)*, Special Issue, 202–204. ISSN:0975-0282, May 2019
15. Jayalakshmi S, Divya L, Vithiya Lakshmi S (2019) Diet monitoring and health analysis using artificial intelligence. Special Issue Published in *Int J Adv Networking Appl (IJANA)*, Special Issue, 234–237. ISSN:0975-0282, May 2019
16. Sunita MK, Dr Sundari G (2020) A frame for brain tumor segmentation and classification using deep learning algorithm. (IJACSA) *Int J Adv Comput Sci Appl*
17. Lerousseau M, Eric D, Nikos P (2020) Multimodal brain tumor classification. In: *International MICCAI Brainlesion Workshop*. Springer, Cham, pp 475–486
18. Vani N, Sowmya A, Jayamma N (2017) Brain tumor classification using support vector machine. *Int Res J Eng Tech (IRJET)* 4(7):792–796
19. Zaw HT, Noppadol M, Khin YW (2019) Brain tumor detection based on Naïve Bayes Classification. In: *2019 5th International Conference on engineering, applied sciences and technology (ICEAST)*. IEEE, pp 1–4
20. Rajendra T, Ranjan KC, Md. Irfan M, Shibam M, Ganesha M (2020) Brain tumor detection and classification using machine learning. (IJIRSET) *Int J Innov Res Sci Eng Technol* July 2020
21. Sajjad M, Khan S, Muhammad K, Wu W, Ullah A, Baik SW (2019) Multi-grade brain tumor classification using deep CNN with extensive data augmentation. *J Comput sci* 30:174–182
22. Sultan HH, Salem NM, Al-Atabany W (2019) Multi-classification of brain tumor images using deep neural network. *IEEE Access* 7:69215–69225
23. Zulpe N, Vrushsen P (2012) GLCM textural features for brain tumor classification. *Int J Comput Sci Issues (IJCSI)* 9(3):354
24. Seetha J, Selvakumar Raja S (2018) Brain tumor classification using convolutional neural networks. *Biomed Pharmacol J* 11(3):1457
25. Muhammad H, Hesaputra ST, Handayani A, Mengko TR (2018) Brain tumor classification using convolutional neural network, May 2018
26. Mohsen H, El-Sayed ESA, El-Dahshan ESM, Salem M (2018) Classification using deep learning neural networks for brain tumors. *Future Computing Inform J* 3(1):68–71
27. Afshar P, Arash M, Konstantinos NP (2018) Brain tumor type classification via capsule networks. In: *2018 25th IEEE international conference on image processing (ICIP)* IEEE, pp 3129–3133
28. Kumar S, Chetna D, Sunila G (2017) Classification of brain MRI tumor images: a hybrid approach. *Procedia Comput Sci* 122: 510–517
29. Dahab D, Aboul S, Ghoniemy SA, Gamal MS (2012) Automated brain tumor detection and identification using image processing and probabilistic neural network techniques. *Int J Image Processing Vis Commun* 1(2):1–8
30. Togacar M, Burhan E, Zafer C (2020) BrainMRNet: brain tumor detection using magnetic resonance images with a novel convolutional neural network model. *Med Hypotheses* 134:109531
31. Sharma M, Sourabh M (2013) Brain tumor segmentation using genetic algorithm and artificial neural network fuzzy inference system (ANFIS). In: *Advances in computing and information technology*. Springer, Berlin, Heidelberg, pp 329–339

32. Sapra P, Rupinderpal S, Shivani K(2013) Brain tumor detection using neural network. *Int J Sci Modern Eng (IJISME)*. ISSN 2319-6386
33. Serj M, Fatan B, Lavi G, Hoff D, Puig V (2018) A deep convolutional neural network for lung cancer diagnostic. arXiv preprint. [arXiv:1804.08170](https://arxiv.org/abs/1804.08170)
34. Salehinejad H, Shahrokh V, Tim D, Errol C, Joseph B (2018) Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 990–994
35. Zhenghao SX, Lifeng H (2010) Application of neural networks in medical image processing. In: ISBN the Second International Symposium on Networking and Network Security, April 2010

Mining Suitable Symptoms to Identify Disease Using Apriori and NBC



R. Sandha and M. Vasumathy

Abstract The availability of data mining algorithm leads to extraction of helpful knowledge through data analysis. Researchers are concerned with using these algorithms in various fields. In the medical field, data mining tools are providing successful results through disease diagnosis applications. Health care professionals used data mining and statistical tools to diagnose the disease, However, using data mining techniques for identifying frequent symptoms for a particular disease and applying in patient data to check whether they are affected or not using a classification algorithm leads to suitable treatment for the diseased patients. This paper proposes a model to systematically close those gaps using our methods.

Keywords Data mining · Frequent symptoms · Apriori algorithm · NBC association rule

1 Introduction

Data mining is about finding valid, new and likely effective knowledge and coherent pattern from large data with the help of databases. The data mining is an interdisciplinary that refers to several domains likes parallel computing artificial intelligence, machine learning, mathematical statistic, parallel computing and so on.

Data mining is about extracting knowledge from enormous data. Data mining is looking at global patterns and relationships that exist in databases [1]. Data Mining has great potential to explore hidden trends in the database in the field of medicine. Identifying relationships and hidden patterns often goes undeveloped. Data mining algorithms are the solution to this problem. Data mining algorithms include classification, clustering, associations, prediction and so on. Mining association rules is one of the most critical application of data mining. To detect associations among

R. Sandha (✉)
Thiagarajar College, Madurai, Tamil Nadu, India

M. Vasumathy
The American College, Madurai, Tamil Nadu, India

datasets in databases, association rules are utilized. These connections are based on the data elements' co-occurrence [2]. This paper concentrates on analyzing medical data. Medical information such as symptoms of patient can be used in mining the frequent symptoms of the disease and diagnose the disease of patients using the classification algorithm.

2 Review of Literature

Soni et al. [3] published an assessment of current methodologies for database knowledge discovery utilizing data mining techniques, which is employed in the prediction of heart disease [4] Carlos Ordonez. The issue of finding constrained association rules for heart disease prediction was discoursed. Three imperatives were presented to diminish the number of patterns. To begin with, one requires the traits to seem on as it were one side of the run the show. The moment one isolates traits into uninteresting bunches, the extreme limitation confines the number of qualities in a run the show. Tests outlined that the limitations decreased the number of found rules strikingly other than diminishing the running time. Maria-Luiza Antonie, et al. conducted some tumor detection trials in digital mammography. Antonie et al. [5] discussed several methods such as Genetic Algorithms, fuzzy categorization and Neural Networks. Reference [6], the authors looked into some of the difficulties that could be encountered when mining skin data. Bellaachia and Guven [7] describe predicting breast cancer using data mining technique. Subbalakshmi et al. [8] used an information mining displaying technique, specifically Nave Bayes, to construct a Choice Bolster in Heart Illness Forecast Framework (DSHDPS). PCAR: a Productive Mining Affiliation Rules approach present nearly every viable heart attack forecast framework. Deepika et al. [9] describes Data mining in health care management data.

Jaya Rama Krishniah et al. [10] bargains with the outcomes within the field of clustering based on K mean, kNearest Neighbor and Entropy-based cruel clustering algorithms and kmean on the total performance creates known mean based on entropy which is the most excellent opt time for handling information and appears way better in execution of exactness expectation. Rao and Gupta [11] discussed a FIS technique that eliminates the shortcomings of apriori calculations and its time and scanning database is efficient. Stilou et al. [12] linked the a priori method to a database of diabetes patient data and attempted to extract affiliation criteria from the hidden true parameters. The outcomes suggest that the technique used can be quite beneficial to the demonstrative strategy, especially when large amounts of data are involved.

Kaur and Wasan [13] examined the usage of allocation based information mining methods such as decision tree, Rule based and Artificial Neural Network to Arrange to gigantic amount of health related information. Particularly, they consider utilizing classification of data procedures on health issue information. Chao and Defu [14] elaborated on the method of e-commerce user information mining within the setting

of huge information. At last, the Apriori algorithm is connected to mine e-commerce client information.

Wang and Jiang [15] proposed that the Apriori algorithm is improved by using the MapReduce model of Hadoop platform to parallelize processing, and the experimental results show that the improved Apriori algorithm has high efficiency and good stability in big data processing. The above given audit of writing explained the change within the classification, increment within the expectation of breast cancer, heart illness and skin infection. Digital therapeutic data explains a strategy for recognizable proof of the recurrence of different infections in a specific geological range at a specific time as discussed in this paper.

3 Association Rule Mining

Association Mining is an important data mining functionality and so many research studies have been done by researchers. Association rules framing is the core of datamining is used in transactions between items for sales purpose. The usage of these rules is finding unknown relationships, and producing output which can perform the basis for prediction and decision making. In the event of choosing association rules from the given data of all conceivable rules, imperatives on different significance measures are utilized. The most known limitations are least limits on confidence and support. Support percentage is calculated based on the frequency of the item set that appears in the dataset. Support is an indicator of how repeatedly the items appears in the dataset. The support(p) is an item set p is outlined because of the proportion acquired for transactions within the knowledge set that was included in the category set. For example,

The item set Fan, AC, TV gain $5/15 = 0.33$ support in the database because all exchanges achieve 33%. To be more specific, 5 denotes the number of database transactions containing the category set “Fan, AC, TV,” whereas 15 represents the total transactions. The degree of confidence in a rule indicates how frequently it has been proven to be true. Confidence can be utilized for an appraisal of the likelihood. The disclosure of association rules is partitioned into two stages, where the location of the frequent item sets is to begin with stage and era of affiliation rules which is the second stage. Every set of items is called an itemset in the first phase, and a minimum support threshold is given as input. If they appear together more frequently than the minimal support threshold, this itemset is referred to as a frequent itemset.

This stage is more significant than the second since finding frequent item sets is straightforward but expensive. It can currently generate many rules from a single itemset. As a result, all of the rules’ confidence and support criteria should be connected in order to prune the rules whose values fall below the thresholds. Finding the link between several elements from a large number of transactions productivity is a problem that has been addressed by association mining.

For mining frequent itemsets, the apriori algorithm is the most well-known and imperative approach. Apriori is utilized to find all frequently occurring sets in a

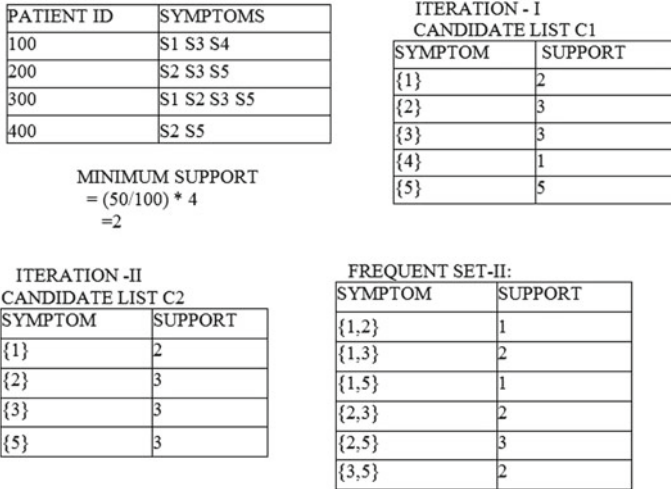


Fig. 1 Support count

database (Fig 1). As per Apriori algorithm frequent itemset subset must moreover be visited. The key thought of Apriori principle is to form numerous iterations over the database collections. It makes use of an iterative approach called breadth first search through the database, where n-itemsets are used to investigate (n+1)-itemsets. Within the starting, the collection of the frequent itemset is found. The set of that contains something which fulfills the support count. In each consequent iteration, we start with a given set of item sets found to be expansive within the past. The given seed set is utilized for producing unused possibly huge datasets, named as candidate item sets, and the real support for those candidate data sets amid the iteration over the information is checked. We finally came to the conclusion as to which among the candidate datasets are really huge, and they have to be the input for another at the conclusion of pass.

4 Proposed Work

The main goal of our work is to develop a model for disease prediction and prevention system using Naive Bayes and apriori. The model will identify the exact symptoms of the particular disease from historical data of different patients using maximum frequent set and find if the exact disease is affecting the patient or not using classification algorithm. It may reply intricate inquiries for detecting illness and hence help medical professionals to create shrewd decisions clinically which cannot be done by conventional bolster frameworks. By giving successful medicines, it moreover makes a difference to reduce the treatment costs. Here, the scope of the project is

that integration of finding the maximum frequent set from patient records and decision support with the classification algorithm might diminish restorative blunders, improve quiet security and diminish the undesirable hone variety.

Information mining devices have been created for compelling investigation of therapeutic data to assist the clinical user in making a proper conclusion. In this research, the researcher can gather information from the clinic which has the adequate points of interest of the patient's side effects. Once the information is collected from the clinic data framework, research work discovers the frequent symptoms of an infection with the assistance of association techniques. Using frequent symptoms of a disease, the affected people can easily be identified using the decision tree algorithm which reduces the diagnose time for fast spreading viruses like the corona that affected a huge crowd of people.

This research helps in mining the information about frequent symptoms of an illness with the use of data from the training set and find the affected people using naïve Bayes algorithm. The clinic information system is used in the medical profession to obtain various types of treatment databases of illnesses and the affected persons from various locations. It may be a herculean errand to recognize the frequent diseases and its causes from the expansive dataset. Affected people from diverse locations approach different centers for healing. Their data are maintained by the clinics where they get their treatment. Collecting information around the regularly occurring ailments is not a straightforward work. The information collection with respect to these types of indications of an illness can be done with the help of the association rule. Particulars with respect to these types of diseases in a particular time can too be mined utilizing the apriori algorithm.

4.1 Apriori Algorithm

```

Ci: dataset of size i
Li: frequent symptom set of size i L1= {frequent set};
For (i= 1; Li!=0; i++)
do begin Ci+1= Medical data generated from Li;
each transaction t in database
does the count increment of all
medical data in Ci+1that are
contained in t Li+1= medical data
in Ci+1with min_support end
return; Li;

```

The proposed system is useful to differentiate the frequent disease or illness symptoms in a massive restorative information set. The end result of this is to inquire if assistance can be offered to the professionals in developing restorative options for the different health issues.

ITERATION -III CANDIDATE LIST C3	
SYMPTOM	SUPPORT
{1,3}	2
{2,3}	2
{2,5}	3
{3,5}	2

FREQUENT SET	
SYMPTOM	SUPPORT
{1,2,3}	1
{1,2,5}	1
{1,3,5}	1
{2,3,5}	2

SYMPTOM	SUPPORT
{2,3,5}	2

ASSOCIATION RULE	SUPPORT	CONFIDENCE	CONFIDENCE %
2 -> 3	2	2/3=0.66	66%
3 -> 5	2	2/3=0.66	66%

Fig. 2 Confidence

Minimum confidence is 50%. So, both the rule outputs as the confidence above 50% is shown in Fig 2. The final rules are,

Rule 1 : 2 -> 3

Rule 2 : 3 -> 5

Maximum frequency of symptoms identified from the given symptoms through the apriori algorithm. It is considered as an attribute for the NBC classifier to find the highest probability of symptoms which lead to a correct decision about the disease.

NBC may be a classification procedure based on a suspicion of freedom between interpreters. In straightforward terms, a NB classifier expected to be the nearness of a specific highlight in a lesson is irrelevant to the existence of any other feature. To construct a Bayesian demonstration is basic and especially functional in the event of massive data sets. Besides straightforwardness, NB classifier is known to be at convenient classification strategies as well. Bayes theorem provided a better way of finding back likelihood $Pro(c|x)$ from $Pro(x)$, $Pro(x|c)$ and $Pro(c)$. $Pro(c|x)$ is the posterior probability of the attribute given the indicator (attribute). The prior probability of the class is $Pro(c)$. Probability of the predictor given in the class is $Pro(x|c)$. The very first probability of the predictor is $Pro(x)$. Let's follow the below steps to find it.

Make a decision as per Fig 3.

Step 1: Change over the information fix to the recurrence table Fig 4.

Step 2: To find the probabilities, make a Probability table.

Fig. 3 Decision table

Patient	Result
P1	Yes
P2	No
P1	Yes
P1	Yes
P2	No
P2	No
P3	No
P3	No
P3	No
P1	No
P1	No
P2	No
P3	Yes
P3	Yes

Fig. 4 Frequency table

Frequency table

	No	Yes
P1	2	3
P2		4
P3	3	2

Likelihood

	No	Yes	
P1	2	3	0.36
P2		4	0.29
P3	3	2	0.36
	0.36	0.64	

Step 3: Presently, use the NBC equation to calculate the back likelihood for each lesson. The lesson with the most useless back likelihood is the solution of the prediction.

Using the above discussed method we can solve it, so $Pro(Yes | P1) = Pro(Yes)/Pro(P1) * Pro(P1 | Yes)$. Here, we have $Pro(Yes) = 9/14 = 0.64$, $Pro(P1) = 5/14 = 0.36$, $Pro(P1 | Yes) = 3/9 = 0.33$ Now, $Pro(Yes | P1) = 0.330 * 0.640/0.360 = 0.60$, which has a larger probability. Naive Bayes classifier employs a comparative strategy to anticipate the likelihood of diverse course based on different traits. This technique is generally utilized in content classification and with issues having numerous classes. From these observations Sensitivity, Accuracy and Specificity can be characterized as follows.

Accuracy: Accuracy is Eq. 1. The number of true positives is added with number of true negatives divided by the total of number of true positives, true negatives, false positives and false negatives

$$\text{Accuracy} = \frac{\text{number of true positives} + \text{number of true negatives}}{\text{number of true positives} + \text{false positives} + \text{false negatives} + \text{true negatives}} \quad (1)$$

Sensitivity: Sensitivity is Eq. 2. The number of true positives is divided by the number of true positives and number of false negatives

$$\text{Sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}} \quad (2)$$

Precision: precision is Eq. 3. It is defined as the rate of the true positives against both false positives and true positives.

$$\text{Precision} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{false positives}} \quad (3)$$

Specificity: Specificity is Eq. 4. The number of true negative rate divide by both the number of true negatives and true negatives and false positives.

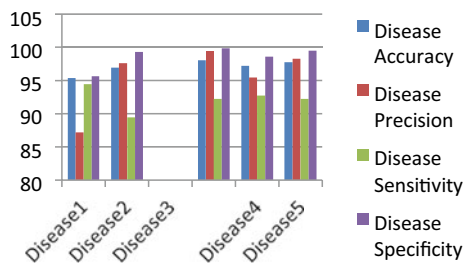
$$\text{Specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}} \quad (4)$$

In Table 1. Naïve Bayes classification made based on Accuracy, Precision, sensitivity and specificity. Five diseases were taken for research and it produces a higher percentage of result.

Based on Fig 5. Disease 3 gets the highest accuracy than other diseases. Disease 3 gets the highest precision than other diseases. Disease 1 gets the highest sensitivity than other diseases, whereas Disease 3 gets the highest specificity than other diseases.

Table 1 Naive Bayes

	Accuracy	Precision	Sensitivity	Specificity
Disease1	95.33	87.17	94.44	95.62
Disease2	96.93	97.57	89.44	99.29
Disease3	98.002	99.4	92.22	99.82
Disease4	97.203	95.42	92.77	98.59
Disease5	97.73	98.22	92.22	99.47

Fig. 5 Classification

5 Conclusion

This research proposes the association rule-based apriori information mining procedure which can find the occurrence of disease side effects and influencing patients who will recognize utilizing NBC. This research study is done on different patients from different topographical areas and at different times. A data mining technique is used to compute the frequency of disease symptoms, which is then used in a classifier to generate a decision table to locate persons who are affected. It examined different diseases and was validated using metrics. The analysis done was the actual fact that time consumption is taking place for identifying symptoms of a disease and for recovery. Data obtained from different hospitals are used as working data set for training the data. Completely, 1000 quiet records influenced by diverse symptoms amid the year 2020 are analyzed. WEKA information mining apparatus is utilized to distinguish the recurrence of the infections that are repeating in individuals located in different geological areas amid diverse time periods.

References

1. Pujari AK (2001) Data mining techniques. Edition 2001
2. Ranjan J (2007) Applications of data mining techniques in pharmaceutical industry. *J Theor Appl Inf Technol*
3. Soni J et al (Mar 2011) Predictive Data mining for medical diagnosis: an overview of heart disease prediction. *Int J Comput Appl* (0975–8887) 17(8)
4. Ordonez C (2004) Improving heart disease prediction using constrained association rules. Seminar Presentation at University of Tokyo
5. Antonie M-L et al (2001) Application of data mining techniques for medical image classification. In: Proceedings of the second international workshop on multimedia Data Mining (MDM/KDD'2001), in conjunction with ACM SIGKDD conference. San Francisco, USA, August 26, 2001
6. Barati E et al (2011) A survey on utilization of data mining approaches for dermatological(Skin) diseases prediction. *Cyber J*
7. Bellaachia A, Guven E (2006) Predicting breast cancer survivability using data mining techniques
8. Subbalakshmi G et al (2011) Decision support in heart disease prediction system using Naive Bayes. *Indian J Comput Sci Eng (IJCSSE)*

9. Deepika N et al (2011) Association rule for classification of Heart-attack patients. *Int J Adv Eng Sci Technol* 11(2):253–257
10. Jaya Rama Krishniah VV et al (May 2012) Predicting the heart attack symptoms using biomedical data mining techniques. *Int J Comput Sci Appl (TIJCSA)* 1(3). ISSN: 2278-1080
11. Rao S, Gupta P (Jan–Mar 2012) Implementing improved algorithm over APRIORI data mining association rule algorithm. *Int J Comput Sci Tech nol (IJCST)* 3(1). ISSN: 0976-8491(Online)ISSN: 2229- 4333 (Print).
12. Stilou S, Bamidis PD, Maglaveras N, Pappas C (2001) Mining association rules from clinical databases: an intelligent diagnostic process in healthcare. *Stud Health Technol Inform* 84(Pt 2):1399–1403
13. Kaur H, Wasan SK (2006) Empirical study on applications of data mining techniques in healthcare. *J Comput Sci* 2(2):194–200
14. Chao X, Defu H (2020) Application of apriori algorithm in user data mining in electronic commerce, vol 114. Springer, AISC. 08 May
15. Wang H, Jiang H, Wang H (18 Nov 2020) Research on an improved algorithm of Apri ori based on Hadoop. *IEEE Xplore*

Background Features-Based Novel Visual Ego-Motion Estimation



B. Sharmila and D. Nedumaran 

Abstract For localization in mobile robotics, usage of camera provides hundreds and thousands of information about its environment and poses a great challenge computationally in real-time applications. Furthermore, when it is applied in computer vision algorithms, its robustness turns out to be a vital factor. This paper contributes a robust and efficient visual Ego-motion estimation technique using a stream of images captured solely in a stereo camera. In many cases, nuisances caused due to instantly varying illumination, occlusions, and low textured regions that increase the error rate of tracking visual cues. A linear multi-orientation, multi-scale image decomposable steerable pyramid transformation was applied to images to overcome the above mentioned problems. Using the Steerable pyramid transform, depth of the image features was measured, features closer to camera were neglected based on the threshold value and pure background was extracted for further processing. This step enhances the speed due to less memory consumption, as most reliable feature vectors are commonly present at the static objects of the scene (background). Using Zero Normalized Cross Correlation, singular points containing reliable feature points were extracted. Further, the RANSAC method was used to eliminate the outliers effectively. Finally, Affine transformation was employed to derive the pure rotational and translational values of the camera. The developed algorithm was tested on different stereo sequences and the results were compared with its ground truth values. The results show a good accuracy and achieved 1.09% translational error with 0.027 deg/m rotational error at a speed of 1.96 s.

Keywords Ego-motion · SPT · ZNCC · Stereo camera · RANSAC

B. Sharmila · D. Nedumaran (✉)

Central Instrumentation and Service Laboratory, University of Madras, Chennai, Tamil Nadu, India

e-mail: dnmaran@gmail.com

1 Introduction

The capability of perceiving an unmanned vehicle or robotic environment is directly linked to its increased autonomy. It could be enabled to use these environmental perceptions to elaborate its control and localize itself. In the domain of computer vision, Longuet-Higgins and Prazdny [1] developed the first computational model for the self-motion of the observer, known as Ego-motion. The process of estimating a camera's or observers' instantaneous displacement from a visual input is known as visual Ego-motion estimation. Some of its major real-time applications are robotic navigation [2, 3], virtual reality [4], self-governing or semi-autonomous driving [5]. A great advantage of estimating Ego-motion using visual input is to avoid any external physical device on the system that is already equipped with a vision sensor (camera) for control, observation or any other reason, resulting in the reduction of weight, space, and money. Conventionally, these external sensors have their limitations: wheeled odometry are unsuccessful in smooth or pitchy landscape, while inertial sensors are unreliable due to error accumulation over time, and GPS won't be available in deep indoors or in low coverage areas. Cameras do have another advantage of providing visual information very similar to that of human vision, especially stereo cameras. In spite of having many benefits, vision sensors while used as a single source of information inherit their own noise due to changes in illumination, contrast, difference in texture, occlusion etc. These drawbacks make the tracking of complex features into a very difficult task. Considering all these facts, the Ego-motion estimation in a dense background is a complex and challenging task.

In deriving a solution for the above issues, a novel Ego-motion computation method based on stereo vision is presented. Linear multi-orientation and multi-scale image decomposition was attempted by steerable pyramid transform (SPT) to eliminate the errors due to noise, illumination changes, and texture variation in the sequence of input images. This step helped in separating the image information from noise and enhancing the tracking of complex image features mainly through the decomposition of the image and reduction in memory bandwidth that enables a decrease in computational time. SPT was chosen, owing to its unique nature of translational and rotational invariant features, which simultaneously decreases the image size, removes the noise, and improves the feature properties present in the scene. Further, depth information of image features were derived from SPT processed stereo image pairs. By introducing a threshold value the foreground information was filtered out (i.e., independently moving objects) from background consisting of static objects with the most valuable information. This step further accelerates the process, providing a way to implement our algorithm in real-time applications. Subsequently, the Zero Normalized Cross Correlation (ZNCC) method was implemented to detect the similarity between singularity points. This method allows us to distinguish reliable feature points from other pixels and also checks the confidence of the detected matches between and frame. After the estimation of the motion vector fields, a RANSAC (RANDOM SAMPLE CONSENSUS) algorithm was applied to divide the data into inliers and outliers to yield a minimal set of inliers having greatest

support. Finally, an affine transformation for inliers was employed to achieve the desired rotational and translational information of the camera's Ego-motion. The key objective of this paper is to develop a real-time system with high speed and less computational complexity, by using the proposed simple and efficient techniques. The results of this study revealed a high precision and good accuracy in outdoor terrain, which is ascertained in terms of an average translational error value of 1.09% and rotational error value of 0.027 deg/m at a speed of 1.96 s. The advantage of our method mainly depends on reducing the size of the image data and processing time, which paves the way to implement the Ego-estimation in hardware environment like FPGA, GPU, and DSP, for further improvements in the speed of Ego-motion estimation.

1.1 Related Work

Because of its potential applications in real-time robotic control domain, visual odometry has an increased attention over past few years. In the field of visual odometry a.k.a. Ego-motion estimation, a great deal of work was conducted by Nister et al. [6]. Traditional methods for estimating the observer's Ego-motion includes more than one sensor assembly. For example, GPS, IMU, IR imaging systems, etc. But, the influence of sensor errors on motion estimation caused due to mechanical issues like wheel slipping in slippery terrains or accuracy and occasional absence of GPS in remote areas affects the precision of the Ego-motion estimation. Subsequently, vision data was used alongside of mechanical sensors that introduced a new hybrid system with active and passive sensors. Considering a flying insect vision system as a model, using optical flow fields derived from vision sensor and data from IMU, an opto-aeronautic algorithm was developed to estimate the Ego-motion [7]. As synchronizing those sensors resulted in error accumulation over time, this method lacked in accuracy and also consumed more power and energy. Many other works describing the hybrid system were reported in detail [8–11]. Scaramuzza and Fraundorfer [12] explained that Ego-motion estimation with vision sensors usually leads to very small relative position error ranging between 0.1 and 2% of the true observer's position. In 2007 [13, 14], the usage of visual Ego-motion estimation method in Mars exploration rovers was implemented which also demonstrated the efficiency of vision system on a different planet. Generally, cameras are present in almost every computer vision/automated/robotic systems. Using the data available on the system, instead of incorporating a new one, vision-based estimation proves as a cost effective and valuable error-minimizing technology. Therefore, Ego-motion estimation is a potential tool for robust real-time applications. With lot more advantages than mechanical sensors, vision sensors have their own limitations due to lightning conditions, unknown distance of the objects from camera, texture variation, blurring, etc. Usage of different types of vision sensors solves these limitations. Monocular, Omni-directional, stereo, compound eye cameras are the most used camera models. Self-motion estimation using wide field of view camera and insect eye inspired

compound eye camera were briefly explained in [15, 16]. Applications involving monocular camera poses problems like unusual viewing conditions such as camera flipping vertically and absence of visible ground plane. Objects moving independently can lead to valid motion vectors, which are nonetheless outliers. In addition to this, forward motion: a very common problem in real-world autonomous navigation, is particularly difficult for monocular Ego-motion estimation [17]. Attempting Ego-motion estimation using a single camera inherits the problem of determining the range of an object as well as definite ambiguities that occur due to small rotational and translational motion. Stereo vision camera gives a firm solution for all the above-mentioned monocular vision problems. These cameras have an advantage of calculating distance of the objects in simpler way with good accuracy and speed [18]. Defining an appropriate baseline (distance between two center points of the cameras in stereo setup) gives us some valuable in-depth information of the scene. Initially, to minimize data size of the input image sequence without affecting the quality, Steerable Pyramid Transformation (SPT) [19] was applied. A scene consisting of uneven texture patterns and brightness shot by stereo camera is difficult to decide whether pixels at the left and right images correspond to one another. This leads to an unauthentic match of calculating the disparity map. SPT developed by Freeman and Adelson [20] was tested for scene motion estimation [21], recognition of objects [22], and stereopsis [23]. As SPT has a tendency to produce feature descriptors invariant to both rotation and translation, we have used it as a front end of our algorithm. Optical flow is a dense motion algorithm used to track the motion of an image with brightness patterns [24]. Ego-motion based on optical flow fields were studied in [25–27]. In optical flow fields, moving sources of light with respect to the observer affected their subsequent stability of robustness.

In recent years, feature-based Ego-motion estimation is gaining momentum. The key point for an efficient motion estimation algorithm depends on the selection of features, which play a critical factor for best error-free results with good accuracy and speed. The existing methods of estimating motion between two scenes can be classified as feature based [28–31], appearance based [32], and pixel based [33]. Szeliski surveyed the effectiveness of feature based and pixel based methods in selecting the feature points [34]. In [35], the authors briefly evaluated some basic and commonly used feature detectors. Common approaches use local feature detectors such as scale-invariant feature transform (SIFT) [36], Speeded up Robust Features (SURF) [37], Features from Accelerated Segment Test (FAST) [38], ORB (Oriented FAST and Rotated BRIEF) [39] or in some special cases are even combined together [40] at the cost of higher computational complexity. Systems associated with the Kalman filter framework [41–43] tracks the image sequence with respect to the detected features and predicts the future camera state. This system tends to fail over long distances, as the usage of large number of features increases the computational cost. Another method of feature tracking is the structure from motion (SFM), which estimates the relative position of more than two cameras with respect to their frames by matching them [6, 44–47]. SLAM (Simultaneous Localization and Mapping) [48] estimates Ego-motion by simultaneously updating the map of its surroundings. Besl and McKay implemented ICP (Iterative Closet Points) for finding the rotation and

translation between two point clouds by shortening the geometric difference between them [49]. Further, the extension of ICP with Least Trimmed Squares (LTS) [50] was experimented. In this work, ZNCC (Zero-mean Normalized Cross Correlation) [51, 52] was used as the similarity measure between detected corners of background features using the Harris corner detector [53]. Subsequently, outliers were reduced by building a model parameter set with Random Sample Consensus (RANSAC) [54–57].

The paper is structured as follows: Sect. 2 briefly discusses the flow chart of the proposed method along with the strategy of selecting reliable feature points and their basic algorithms used in this work. Section 3 has elaborated details of the methods and tools used to implement the proposed method. In Sect. 4, the obtained results were studied and Sect. 5 covers the concluding remarks of the proposed work with future scope.

2 Algorithm

2.1 System Overview

Figure 1 shows the block diagram of the proposed algorithm. Using the stereo imaging arrangement, the video sequences from the left and right sides were acquired for Ego-motion estimation. Most importantly, Ego-motion estimation of a camera mainly relies on the choice of feature extraction, which always focuses on speed, accuracy and repeatability. The traditional methods often fail to consider the memory occupied by the frames of the input video sequence and the intermediate images, in addition to the complicated computational algorithms.

The proposed method differs from other methods in terms of efficient handling of the memory constrain problem without compromising the speed and accuracy of the system. The choice of SPT for base operation not only helps to remove unwanted noise, illumination issues, texture variation etc., but also reduces the input image size without affecting the quality of the data and preserves the feature details. This step boosts the speed of the system in a remarkable way for real-time implementation without compromising its accuracy and stability. Then, SPT was applied to both left and right images and a disparity map was constructed. The resulting disparity map was applied with a threshold of. At this stage, the background of the image sequence was extracted, which further decreases the complexity of applying feature extraction algorithm for the entire image, since the reliable singularity points are often present at the background. To extract the pure background, comparison of the right image with the threshold $T_{sh} < 25$ was performed. Further, Harris corner detection was applied between the K and $K + 1$ frames and matched to identify the exact feature points. Additionally, ZNCC was applied to eliminate erroneous data by way of checking the confidence of the match. The extracted reliable motion vectors from the ZNCC are then subjected to RANSAC algorithm for removing the outliers,

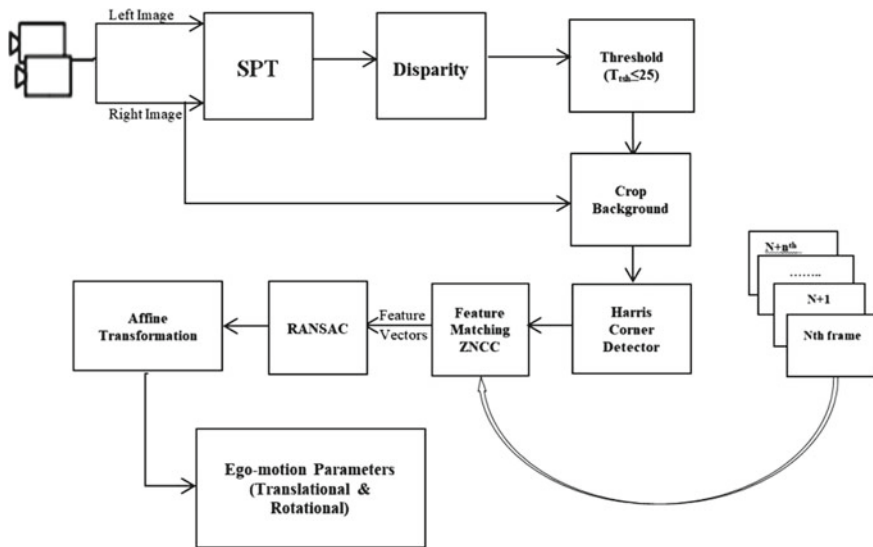


Fig. 1 Block diagram of the proposed method

which build small sets of vectors for constructing a testing hypothesis. The hypothesis with the maximum score was considered as the best solution having the translational and rotational information of the camera. In the final step, the best hypothesis was interpolated using Affine transformation [58] for achieving the optimal Ego-motion parameters.

2.2 Steerable Pyramid Transformation (SPT)

SPT was used in this work as the front-end for basic pre-processing and stereo disparity identification from the frames of the input video sequence. Steerable pyramids are familiar in evaluating the image in several orientations at multiple scales of resolution. When a steerable filter output was convolved with an image at different orientation, it seems to be loaded with information about local neighborhood of each pixel [59]. Each level of Pyramid filtering includes information about both the space and the spatial frequency [60]. Thus, the steerable pyramid produces feature vectors that are translational and rotation invariant, owing to the advantage of orthonormal transform characteristics. Let, I_l and I_r be the left and right images at time t . Pyramid formation consists of two steps: analysis and synthesis. Step 1: In analysis part I_l and I_r were partitioned as low pass sub-band (with steerable filter L_0) and high pass sub-band (with steerable filter H_0). Successive level of steerable pyramid was built up from the preceding step by convolving the low-pass filter L_1 with the set of



Fig. 2 Four basis filters oriented at 0° , 45° , 90° and 135°

low-pass sub-bands (B_1, \dots, B_k). For j th order filter, the basic filter functions needed for steering appears till $j + 1$ orientation. Figure 2 illustrates a first level Steerable pyramid decomposition. The shaded area over the lower low-pass sub-band was sub-sampled by 2 in x and y directions.

Further, the steerable filters L_0 and H_0 defined in the Fourier domain [61, 62] are given in Eqs. (1) and (2).

Where are polar frequency co-ordinates of the filters? A third-order filter set requires four different basic filters oriented at and. SPT provides finer edge features at higher spatial frequency sub-bands, while typical edge features are incorporated in the lower spatial frequency sub-bands [62]. A steerable basis set was obtained by using as basis filter, which has 4 different orientations (shown in Fig. 2). The linear combinations of the basis filter forms the coefficient of the filter orientation set. In this work, a 3-level single orientation SPT was implemented. This step reduces the size of the input image frames without losing key features of the image and also removes the unwanted noise data.

$$L_0(r, \theta) = \begin{cases} 1 & \text{if } r \leq \frac{\pi}{4} \\ \cos\left(\frac{\pi}{2} \log_2\left(\frac{2r}{\pi}\right)\right) & \text{if } \frac{\pi}{4} \leq r \leq \frac{\pi}{2} \\ 0 & \text{if } r \geq \frac{\pi}{2} \end{cases} \quad (1)$$

$$H_0(r, \theta) = \begin{cases} 0 & \text{if } r \leq \frac{\pi}{4} \\ \cos\left(\frac{\pi}{2} \log_2\left(\frac{r}{\pi}\right)\right) & \text{if } \frac{\pi}{4} \leq r \leq \frac{\pi}{2} \\ 1 & \text{if } r \geq \frac{\pi}{2} \end{cases} \quad (2)$$

From I_L and I_R , feature vectors $F_L(x, y)$ and $F_R(x, y)$ were derived by combining their steerable pyramid responses at each level, respectively. These vectors are enriched with valuable description about their intensities in the image. Later, a disparity map was constructed by minimizing these feature vectors of left and right images using the Mean Square Error (MSE) [63] given by Eq. 3.

$$MSE = \frac{1}{F_s} \sum_{i \in F_s} (F_L(x, y) - F_R(x, y))^2 \quad (3)$$

Here F_s denotes the feature vector size. Window with minimum disparity estimate was taken as the best initial disparity value. Locations, where no possible match was identified (like constant intensity regions) and at boundaries, disparity was assigned

as zero. As a form of smoothness constraint, median filter was applied to the full resulting image frame. The final smoothed image D_s gives the disparity map of stereo image pair I_L and I_R .

Here, the background of input stereo image frame was segmented through thresholding of its disparity map to a required minimum value. This step helps in speeding up the Ego-motion estimation, as the most reliable singular points lie at background information. Discarding the foreground objects simplifies the remaining process as it releases unwanted memory allocation and improves robustness against false estimation. A condition of $D_{th} \leq 50$ was applied to D_s in the proposed method.

2.3 Keypoint Detection and Matching

Once the background was separated, the resulting image was then processed to extract reliable keypoints. Keypoints are defined as any salient point, otherwise known as interest points of an image, often a feature or a corner detail. A corner point has remarkably larger change in intensity from rest of its neighboring image points. Harris corner detector is a tool for finding isotropic saliency metric, proposed by Harris and Stephens [53], was used in this work for the detection of reliable corners or keypoints. In this method, autocorrelation matrix was used to evaluate the approximate autocorrelation function at any arbitrary direction. Considering an image I with pixel value (i, j) at image position $I(i, j)$ having an autocorrelation function of $f(i, j)$, the autocorrelation matrix was defined in Eq. (4) based on few assumptions [53]. In Eq. (4), $\Delta i, \Delta j$ represents some small shift in directions i, j , respectively from the image point $I(i_n, j_n)$.

$$f(i, j) \approx (\Delta i \ \Delta j)A \begin{pmatrix} \Delta i \\ \Delta j \end{pmatrix} \quad (4)$$

$$G_w(i, j; \sigma) = \exp\left(-\left(\frac{i^2 + j^2}{2\sigma^2}\right)\right) \quad (5)$$

$$I_i = \frac{\partial I}{\partial i} = I * (-1, 0, 1); \quad I_j = \frac{\partial I}{\partial j} = I * (-1, 0, 1)^T \quad (6)$$

A Gaussian window stated in Eq. (5) was used for computing first order derivatives given in Eq. (6). Autocorrelation function's principal variation in orthogonal directions results in Eigenvalues λ_1 and λ_2 of autocorrelation matrix A . When both Eigenvalues are high, they represent a highly textured corner point [64]. In case of one higher and other lower Eigenvalues, there was a variance across one direction representing an edge point. These Eigenvalues were used in Harris corner detector to define the saliency of a pixel. 'Cornerness' C was evaluated using Eq. (7).

$$C = \det(A) - k \text{ trace}(A)^2 \quad (7)$$

where,

$$\det(A) = \lambda_1 \lambda_2 \quad (8)$$

$$\text{trace}(A) = \lambda_1 + \lambda_2 \quad (9)$$

The selected keypoints were matched for similarity with successive frames. For selection of keypoints, an improved version of Normalized Cross Correlation (NCC) known as Zero mean Normalized Cross Correlation (ZNCC) was used. Consider T_1 and T_2 as templates of previously detected keypoints in the first and second images with co-ordinates (x_1, y_1) and (x_2, y_2) , respectively. Let the mean intensity of T_1 be $\overline{T_1}$ and T_2 be $\overline{T_2}$. Equation (10) gives the ZNCC values of the detected motion vector \vec{m} of a corner feature point after matching with successive frames. The ZNCC value corresponding to the best similarity match gives the motion vector value of the required background feature for further process.

$$ZNCC = \frac{\sum_{x_1, y_1} (T_1(x_1, y_1) - \overline{T_1})(T_2(x_2 + x_1, y_2 + y_1) - \overline{T_2})}{\sqrt{\sum_{x_1, y_1} (T_1(x_1, y_1) - \overline{T_1})(T_2(x_2 + x_1, y_2 + y_1) - \overline{T_2})^2}} \quad (10)$$

2.4 Ransac

As the obtained feature vectors may consist of invalid feature correspondences, known as outliers. RANSAC was implemented for rejecting them and extract inliers for the Ego-motion estimation. The hypothesis with the maximum score was taken as inliers. Results were then used to estimate parameters for the Affine Transformation (AT) [58] described in Eq. (11). Least Mean Square (LMS) was applied to selected inliers set to obtain AT. The \widehat{AT} will be chosen as the best Affine transform (AT_B) only, if IL_n (number of inliers detected) reaches a condition of $IL_n \geq 50\%$ of \vec{m} 's total's calculated. Otherwise, the whole process will be repeated again. According to Eq. (12), maximum iteration trails are limited [54]. Equation (12) ensures that IL'_n vectors with a probability of P_s are free from outliers, while ε is the highest fraction of outliers.

$$\begin{pmatrix} i^n \\ j^n \\ n \end{pmatrix} = \begin{pmatrix} A & B \\ D & E \\ 0 & 0 \end{pmatrix} \begin{pmatrix} i^{n+1} \\ j^{n+1} \\ 1 \end{pmatrix} + \begin{pmatrix} C \\ F \\ 1 \end{pmatrix} \quad (11)$$

$$\text{no. of iterations} = \frac{\log(1 - P_s)}{\log[1 - (1 - \varepsilon^{IL_n})]} \quad (12)$$

In Eq. (11) A, B, D, E and C, F are rotational R and translational T parameters, respectively, of the camera motion. Let (i^n, j^n) and (i^{n+1}, j^{n+1}) denote the pixel co-ordinates of detected feature vectors in I_L and I_R , respectively.

3 Experimental Results

The proposed method was tested with various stereo image sequences of resolution 1392×512 pixels having cameras (Ego-motion) randomly with different translational and rotational motions. Furthermore, the input was annotated with ground truth values. The algorithm was programmed in C++ under OpenCV environment and executed on 64-bit operating system with 4 GB RAM. Table 1 lists the number of detected reliable corner points at 300, 600 and 900 frames with respect to different frame rates. Even under different frame rates, the proposed method showed a stable estimation of feature points at any instant of time. While executing our algorithm, around 2.5 GB of RAM space was engaged, this clearly exhibits that the proposed SPT method reduces the total memory occupied and thus speeding up the system for real-time implementation. This will pave possible steps to implement a stand-alone Ego-motion estimating system in the FPGA (Field Programmable Gate Arrays) environment. Figure 3a shows the resulting Ego-motion of the vehicle with ground truth values of sample image sequence recorded in urban terrain. It is very clear that the proposed method estimates the Ego-motion of the camera very close to the ground truth values. Figure 3b shows the trajectory of the proposed method along with the ground truth values, SSLAM and SDWO algorithms. The graph explains the robustness and efficiency of the proposed method against the ground truth and other two algorithms. Ego-motion of the vehicle was estimated in 1.96 s. Further, on testing of the algorithm on several frames, the proposed method generates Ego-motion parameters very closely to the ground truth in most frames.

To estimate error in Ego-motion estimation, Average Translation Error (ATE) and Average Rotational Error (ARE) metrics were calculated. The results of this study exhibit an average error of 1.09% translational error with 0.027 deg/m rotational error. Figure 4a explains the rotational error and Fig. 4b explains the translational

Table 1 Detected feature points and inliers with respect to different frame rates of 300, 600 & 900 frames

Frame no	Frame rate (fps)	Detected feature points	% of inliers
300	15	168	91.07
	30	156	92.94
600	15	232	90.94
	30	214	91.12
900	15	136	91.91
	30	122	92.62

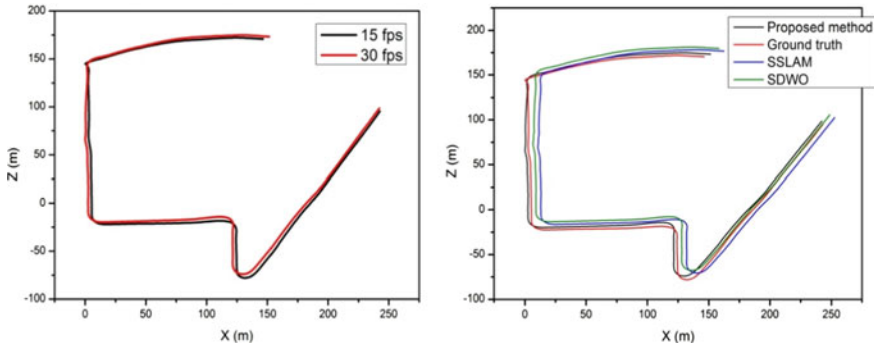


Fig. 3 **a** Estimated Ego-motion of the camera and its respective ground truth values. **b** Trajectory of Ego-motion estimated from the proposed method with ground truth, SSLAM and SDWO

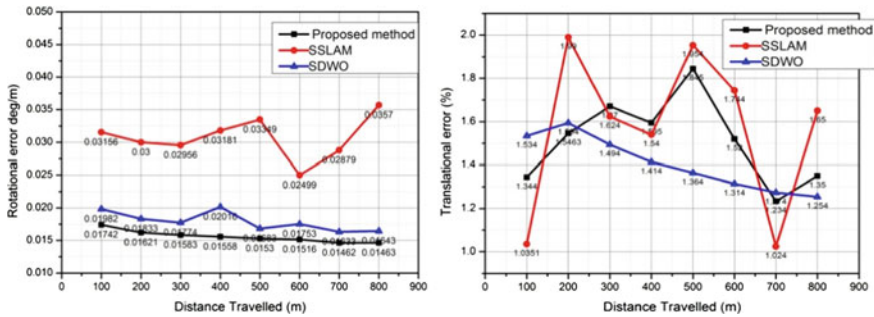


Fig. 4 **a** Rotational error of estimated Ego-motion in sequence_s01; **b** Translational error of estimated Ego-motion on sequence_s01

error for the input image sequence at 15 fps. This ensures that the proposed method improves the speed of Ego-motion estimation without affecting the robustness of the system.

4 Conclusion

In this work, the key property of the Ego-motion of a camera/observer, as transforming the background images into scaled or translated, was used for the Ego-motion estimation of stereo video. The main theme of this work relies on separating the background of an image frame efficiently on most reliable areas of the individual frame and applying feature detector that enabled to speed up the process. The highlight of this work is the contribution of SPT, as it removes noise and defects due to illumination changes without additional processing time, but decreased memory usage with high accuracy. Further, the results demonstrate the number crunching capability of our

approach against different frame speeds. Additionally, ZNCC enhanced the robustness of the proposed method by selecting the best similar reliable feature match. The merit of the proposed method can be understood through the better accuracy as well as the average error of 1.09% in translational movement with 0.027 deg/m in rotational movement at a speed of 1.96 s. In the future scope of this study, we will implement the proposed method in FPGA hardware environment to further improve the speed and built a fully automated stand-alone embedded system for mobile/autonomous driving, video stabilization, motion control and many more in computer vision-based applications.

Acknowledgements The authors are grateful to the Department of Science and Technology for the award of a DST- INSPIRE Fellowship to carry out this research work.

Conflict of Interest The authors hereby declare that there are no conflicts of interest.

References

1. Longuet-Higgins HC, Prazdny K (1980) The interpretation of a moving retinal image. *Proc R Soc London Ser B Biol Sci* 208:385–397
2. Meier L, Tanskanen P, Heng L, Lee GH, Fraundorfer F, Pollefeys M (2012) PIXHAWK: a micro aerial vehicle design for autonomous flight using onboard computer vision. *Auton. Robots* 231855:22–39
3. Stelzer A, Hirschmüller H, Görner M (2012) Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *Int. J. Rob. Res.* 31:381–402
4. Miksik O, Torr PHS, Vineet V, Lidegaard M, Prasaath R, Nießner M, Golodetz S, Hicks SL, Pérez P, Izadi S (2015) The semantic paintbrush: interactive 3D mapping and recognition in large outdoor spaces. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems '15*. pp 3317–3326
5. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the KITTI vision benchmark suite. In: *Proceedings of IEEE conference on computer vision and pattern recognition*. pp 3354–3361
6. Nister D, Naroditsky O, Bergen J (2004) Visual odometry. *Comput. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004*, vol. 1
7. Rutkowski AJ, Miller MM, Quinn RD, Willis MA (2011) Egomotion estimation with optic flow and air velocity sensors. *Biol Cybern* 104:351–367
8. Veth MJ (2006) Fusion of imaging and inertial sensors for navigation
9. Wei L, Cappelle C, Ruichek Y, Zann F (2011) GPS and stereovision-based visual odometry: application to urban scene mapping and intelligent vehicle localization. *Int. J. Veh. Technol.* 2011
10. Tsotsos K, Pretto A, Soatto S (2012) Visual-inertial ego-motion estimation for humanoid platforms. In: *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. pp 704–711
11. Oskiper T, Zhu Z, Samarasekera S, Rakesh Kumar: Visual odometry system using multiple stereo cameras and inertial measurement unit. In: *IEEE conference on computer vision and pattern recognition*. pp 1–8
12. Scaramuzza D, Fraundorfer F (2011) Tutorial: Visual odometry. *IEEE Robot Autom Mag* 18:80–92

13. Bibuli M, Caccia M, Lapierre L (2007) Path-following algorithms and experiments for an autonomous surface vehicle. *IFAC Proc* 40:81–86
14. Cheng Y, Maimone M, Matthies L (2005) Visual odometry on the mars exploration rovers. In: 2005 IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp 903–910
15. Davison AJ, Cid YG, Kita N (2004) Real-time 3D SLAM with wide-angle vision. *Proc IFAC/EURON Symp Intell Auton Veh* 37:868–873
16. Favaro P, Jin H, Soatto S (2003) A semi-direct approach to structure from motion. *Vis Comput* 19:377–394
17. Oliensis J (2005) The least-squares error for structure from infinitesimal motion. *Int J Comput Vis* 61:259–299
18. Diebel J, Reutersward K, Thrun S, Davis J, Gupta R (2004) Simultaneous localization and mapping with active stereo vision. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566). Vol. 4, pp 3436–3443
19. Simoncelli EP, Freeman WT (1995) The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *Proceedings international conference on image processing*. Vol. 3, pp 444–447
20. Freeman WT, Adelson EH (1991) The design and use of steerable filters
21. Huang C, Chen Y (1995) Motion estimation method using a 3D steerable filter. *Image Vis Comput* 13:21–32
22. Ballard DH (1991) Animate vision. *Artif Intell* 48:57–86
23. Jones DG, Malik J (1992) A computational framework for determining stereo correspondence from a set of linear spatial filters. In: *European conference on computer vision*. pp 395–410
24. Beauchemin SS, Barron JL (1995) The computation of optical flow. *ACM Comput Surv* 27:433–466
25. Spies H, Scharr H (2001) Accurate optical flow in noisy image sequences. In: *IEEE International Conference on Computer Vision*. pp. 587–592
26. Black MJ, Anandan P (1993) A framework for the robust estimation of optical flow. In: 1993 (4th) *International Conference on Computer Vision*. pp 231–236
27. Wildes RP (1993) On the qualitative structure of temporally visual motion fields. In: *AAAI*. pp 844–849
28. Corke P, Strelow D, Singh S (2004) Omnidirectional visual odometry for a planetary rover. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). IEEE, pp. 4007–4012
29. Lhuillier M (2008) Automatic scene structure and camera motion using a catadioptric system. *Comput Vis Image Underst* 109:186–203
30. Scaramuzza D, Fraundorfer F, Siegwart R (2009) Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In: 2009 IEEE International conference on robotics and automation. pp 4293–4299
31. Tardif J-P, Pavlidis Y, Daniilidis K (2008) Monocular visual odometry in urban environments using an omnidirectional camera. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp 22–26
32. Scaramuzza D, Siegwart R (2008) Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Trans Robot* 24:1015–1026
33. Milella A, Siegwart R (2006) Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In: *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*. p 21
34. Szeliski R (2006) Image alignment and stitching: a tutorial
35. Song Z, Klette R (2013) Robustness of point feature detection. In: *Computer analysis of images and patterns*. 8048 LNCS. pp 91–99
36. Lowe DG (2004) Distinctive image features from scale invariant keypoints. *Int J Comput Vis* 60:91–110
37. Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: *Leonardis A, Bischof H, Pinz A (eds.) European conference on computer vision ECCV 2006. Lecture notes in computer science*. Springer, Berlin, Heidelberg, pp 404–417

38. Trajtkovii M, Hedley M (1998) Fast corner detection. *Image Vis Comput* 16:75–87
39. Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB—an efficient alternative to SIFT or SURF. In: *International conference on computer vision*. pp 2564–2571
40. Valgren C, Lilienthal AJ (2010) SIFT, SURF & seasons: appearance-based long-term localization in outdoor environments. *Rob Auton Syst* 58:149–156
41. Davison AJ (2003) Real-time simultaneous localisation and mapping with a single camera. In: *International Conference on Computer Vision ICCV’03*. pp 1403–1410
42. Davison AJ, Reid ID, Molton ND, Stasse O (2007) MonoSLAM: real-time single camera SLAM. In: *IEEE transactions on pattern analysis and machine intelligence*. Vol. 29, pp 1052–1067
43. Sola J, Monin A, Devy M, Lemaire T (2005) Undelayed initialization in bearing only {SLAM}. In: *Proceedings of the IEEE/RSJ conference on intelligent robots and systems (IROS)*. IEEE, pp 2499–2504
44. Agrawal M, Konolige K, Agarwal M, Konolige K (2006) Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In: *International conference on pattern recognition ICPR’06*. pp 1063–1068
45. Agrawal M, Konolige K (2007) Rough terrain visual odometry. *Proc. Int. Conf. Adv. Robot.* 25:28–30
46. Agrawal M, Konolige K, Bolles RC (2007) Localization and mapping for autonomous navigation in outdoor terrains: a stereo vision approach. In: *2007 IEEE workshop on applications of computer vision (WACV’07)*. pp 7–12
47. Mouragnon E, Lhuillier M, Dhome M, Dekeyser F, Sayd P (2006) Real-time localization and 3D reconstruction. In: *IEEE computer society conference on computer vision and pattern recognition, CVPR’06*. pp 363–370
48. Bailey T, Durrant-Whyte H (2006) Simultaneous localization and mapping (SLAM): part I the essential algorithms. *IEEE Robot Autom Mag* 13:108–117
49. Besl P, McKay N (1992) A method for registration of 3-D shapes
50. Chetverikov D, Svirko D, Stepanov D, Krsek P (2002) The trimmed iterative closest point algorithm. *Object Recognit. Support. User Interact. Serv. Robot.* 3:545–548
51. Huang J, Zhu T, Pan X, Qin L, Peng X, Xiong C, Fang J (2010) A high-efficiency digital image correlation method based on a fast recursive scheme. *Meas Sci Technol* 21:035101
52. Di Stefano L, Mattoccia S, Tombari F (2005) ZNCC-based template matching using bounded partial correlation. *Pattern Recognit Lett* 26:2129–2134
53. Harris C, Stephens M (1988) A combined corner and edge detector. In: *Proceedings Alvey Vision Conference*. pp 23.1–23.6
54. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* 24:381–395
55. Hong S, Ye C (2014) A fast egomotion estimation method based on visual feature tracking and iterative closest point. In: *IEEE International conference on Networking, Sensing and Control (ICNSC)*. pp 114–119
56. Del-Blanco CR, Jaureguizar F, Salgad L, García N (2008) Motion estimation through efficient matching of a reduced number of reliable singular points. In: *Proceeding of real-time image processing*. Vol. 6811, pp 68110N–68110N–12
57. Yang S-W, Wang C-C (2009) Multiple-model RANSAC for ego-motion estimation in highly dynamic environments. In: *2009 IEEE International Conference on Robotics and Automation*. pp 3531–3538
58. Wisetphanichkij S, DeJhan K (2005) Fast fourier transform technique and affine transform estimation-based high precision image registration method. *GESTS Int Trans Comput Sci Eng* 20:179–191
59. Rabie TF, Terzopoulos D (1998) Stereo and color analysis for dynamic obstacle avoidance. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*. pp 245–252
60. Choudhary BK, Sinha NK, Shanker P (2012) Pyramid method in image processing. *Inf Syst Commun* 3:269–273

61. Mahersia H, Hamrouni K (2015) Using multiple steerable filters and Bayesian regularization for facial expression recognition. *Eng Appl Artif Intell* 38:190–202
62. Ndajah P, Kikuchi H, Yukawa M, Watanabe H, Muramatsu S (2014) An investigation on the quality of denoised images. *Int J Circuits Syst Signal Proc* 5:423–434
63. Tomasi C (1994) Good features to track. In: *Proceedings of IEEE conference on computer vision and pattern recognition*. pp 593–600
64. Fanfani M, Bellavia F, Colombo C (2016) Accurate keyframe selection and keypoint tracking for robust visual odometry. *Mach Vis Appl* 27:833–844

Livspecs: Design and Implementation of Smart Specs for Hearing and Visually Challenged Persons



P. K. Prithvi, K. Chandru, Krishnan B. Yashwanth, Fathima M. Shabika, R. Ranjana , and T. Subha 

Abstract Every day there is a child born with disabilities, but unfortunately, it is not in our hands to undo what has already happened. But we could very well help them to make their lives easier. In this proposal, we have designed an eyeglass that could serve a purpose for people with autism, visually and hearing impaired. People with autism could not sense things immediately; the same is the scenario for the visually impaired. They can't sense what's happening in front of them. So, we have designed an eyewear VI-SPECS (Visually Challenged- Specs) that senses the real-world entities using a camera lens by using YOLO V4. YOLO V4 captures the real-world objects followed by text to speech recognition. This will give them commands about what's happening in front of them. Similarly, for the hearing impaired, we have designed efficient specs HE-SPECS (Hearing Impaired- Specs). This glass takes real word sounds as input, converts them into text segments, and displays them on the screen. The screen will be partially visible so that they cannot disturb while they are at any other work. Thus our vision is to help people to overcome their disabilities.

Keywords YOLO V4 · R-CNN · Autism · Computer vision · OpenCV

1 Introduction

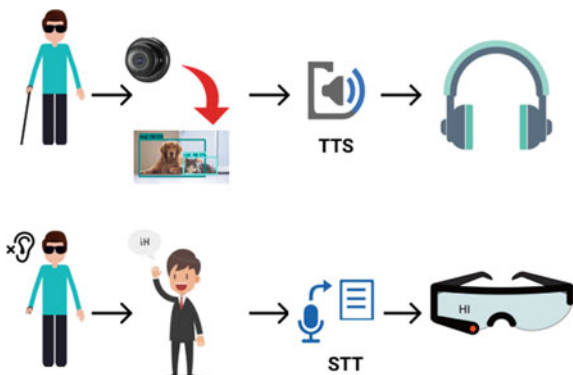
Around 3,85,000 babies are born each day, of which 3% are born with disabilities, which rises to 5% when they are at age one. All these children are not intentionally born or affected by disabilities. In spite of having modern medical assistance, we could detect the disability in the mother's womb. But no assistive technology could

P. K. Prithvi (✉) · K. Chandru · K. B. Yashwanth · F. M. Shabika · R. Ranjana
Sri Sairam Engineering College, Chennai 600044, India
e-mail: sec20it039@sairamtap.edu.in

R. Ranjana
e-mail: ranjana.it@sairam.edu.in

T. Subha
SRM University, Chennai, India

Fig. 1 Framework of proposal



cure or correct the disability of that infant. And in most cases, the disability is found after birth. Neglecting these children is against humanity, and as a parent or well-wisher, one could not see their difficulties. Thus, this is an emotional moment that one could feel that somehow they could help the person or give them any sort of instrument or gadget that will be helpful in an efficient way.

Modern medical technologies couldn't tackle this situation. Hence digital technology came forward to find a solution using emerging technologies and core concepts that helps to assist a person with a disability. YOLO V4 algorithm and most essential functions such as image to speech and speech to text are employed to make our proposed system [1]. The YOLO V4 is an algorithm used for object detection. The use of this algorithm is extensive due to its accuracy. More accurate the prediction, the more the positive outcome. Speech to text and image to speech operations are performed by the third-party control. And the input for these operations is captured in real-time (see Fig. 1).

2 Literature Survey

Garg et al. [2] proposed their idea to improve the accuracy of recognizing the face using aspects of deep learning. In this paper, they used one of the famous deep learning libraries YOLO(YOU ONLY LOOK ONCE). For detecting faces in videos, they used a convolutional neural network as a deep learning approach. They trained and tested the data using the Fddb dataset.

In this paper Xu et al. [3]; They used multifunctional real time electronic glasses (E-glasses) for indoor navigation. The primary functionality of their paper is indoor navigation via navigation algorithm and item of interest recognition via deep learning technology. The resultant object information is given as feedback to the blind person.

Arora et al. [4] developed a paper that detects objects in real-time using image segmentation and CNN (CONVOLUTIONAL NEURAL NETWORK). The position with respect to the person is indicated to the blind individual with the help of speech

stimulus. In this paper they used MobileNetArchitecture along with a single shot multibox detection structure to create a faster and versatile real time object detector.

The paper by Go et al. [5] proposed a methodology to recognize human emotion with the aid of facial images and speech signals. It is accomplished by using multi-resolution analysis from discrete wavelet. And LDA (LINEAR DISCRIMINANT ANALYSIS) generates feature vectors. The emotion recognition from speech signal approach is structured in such a way that the recognition algorithm is performed autonomously.

Bedrii et al. [6] The author's implementation is about automated diet monitoring in unrestrained instances. It is a device which is equipped with a gyroscope and optical sensors that enable it to detect food consumption activities by a person. It captures the images of food intake by an individual with the assistance of an onboard camera.

Önal and Dandil [7] They proposed an idea based on detecting the people in camera where they are not wearing protective suits while working in an industrial environment.

To detect this they used the YOLOv4 deep learning technique. In this study they created their own video dataset and with the aid of YOLO they detected protective equipment like gloves and eyeglasses.

Jung et al. [8] They proposed an idea about a new method for detecting Deep Fakes created by the generative adversarial network (GANs) model by analyzing a major change in the blinking pattern. When eye blinks are repeatedly repeated within a very short period of time, the proposed technique DeepVision is applied as a criterion to validate an abnormality based on the period, repetition number and expired eye blink period.

Camil et al. [8] Their idea is when eye blinks are repeatedly repeated within a very short period of time, the proposed technique DeepVision is applied as a criterion to validate an abnormality based on the period, repetition number and expired eye blink period. Integrity verification can detect deepfakes by observing substantial changes in eye blinking patterns.

Chang et al. [9] Their research project's purpose is to create a prototype method for automatic indexing of sports videos. Their approach is unique in that they suggest combining speech understanding and image analysis techniques to obtain data. In this paper they used image to text classification and text to voice classification.

Rao et al. [10] Electronic Travel Aids (ETAs) aid in navigation by gathering data about the surroundings and conveying it in a manner that enables a blind individual to comprehend the environment's nature. A computer vision-based solution to detecting potholes and uneven surfaces is proposed in order to help blind persons to move easily. The method involves displaying laser patterns, recording them with a monocular video, processing them to get features, and then sending path indications to the blind user.

3 Proposed System

Most of the children in this world are born normal, but on the other hand, some of them are born with some disorders or deficiencies, of which one of the deadliest is Autism. According to a report of the CDC, 1 in 54 children in the US is affected with Autism. There is no leading cause for Autism; it may be due to genetic, non-genetic, or even environmental factors. In our proposal, we have found a way to help children with Autism and visually challenged to make them comfortable when they meet another person or encounter any object in real life. Also, we have planned to design efficient specs that could also serve people with a hearing aid but are visually perfect. This could be a significant relief for them, making them visually connected without eyes. The Backend Architecture Of LIVSPECS is explained in (Fig. 2).

Methodology For VI-Spec (Exclusive specs for visually challenged and child with autism)

To proceed further in this proposal, we have to tackle the following objectives in an efficient way,

- To receive the object motion from the specs through YOLO V4.
- To recognize and process the received data using the Deep Learning model CNN and OpenCV.
- To generate the voice module using the output of the processed information.
- To transfer the voice module through earphones.

Object Detection Using YOLOv4

For image and motion detection, we use smart glasses which comprise in-built speakers to transfer the output, a CPU is incorporated in any one of the arms of the specs which serves as the brain of this gadget, and a tiny camera lens that could

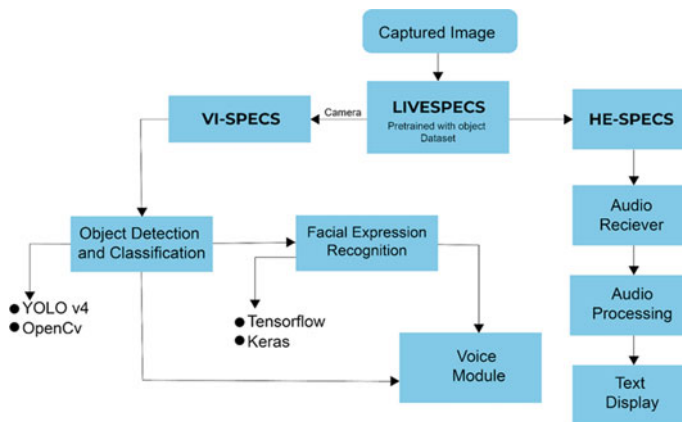


Fig. 2 Backend Architecture Of LIVSPECS

capture the real-world entities is fitted across the lens of the specs. A narrow-angle motion is captured using the camera entity in order to focus on the particular objects that are presented in view with the user. The received motion picture is passed through the YOLOv4 algorithm. A computer vision-based algorithm is developed for rapid and accurate object/image detection using a unique convolutional neural network model. Here image/video is sent as an input to VI-spec. Based on the image sent object detection is done using the you only look once algorithm as mentioned earlier. A grid layer of $S \times S$ size (here S is the number of pixels) is spread across the entire captured image. Instead of detecting the object using layers of neural networks, YOLOv4 focuses on using a grid layer, and outline rectangle/square box across grid layers that outline the entire object residing in the captured image. The proximity of 0 to 1 digit is used for object detection. Where the value 0 is considered as no object is found inside the grid. YOLOv4 also helps in detecting and separating overlapped objects by using the reLU function which operates on $\max(0, S)$ which helps to improve the performance of the software. Thus with the help of YOLOv4 and the python library OpenCV supporting the YOLOv4 algorithm built using CUDA, the motion image is captured and the objects are identified and recognized. The recognized object is then classified based on its properties using OpenCV. I.et the image be classified into different labels such as persons, animals, things, etc., from the dataset provided. The processed output waits for further processing if required or it is automatically transferred to the voice module based on the person using it. Consider the case of a visually challenged person where the output is transferred as the facial expression may not be necessary at times, as they are highly intelligent in understanding intonations (Fig. 3).

To Recognize and Process The Received Data Using The Deep Learning Model CNN and FER

See Fig. 4.

Fig. 3 Facial detection

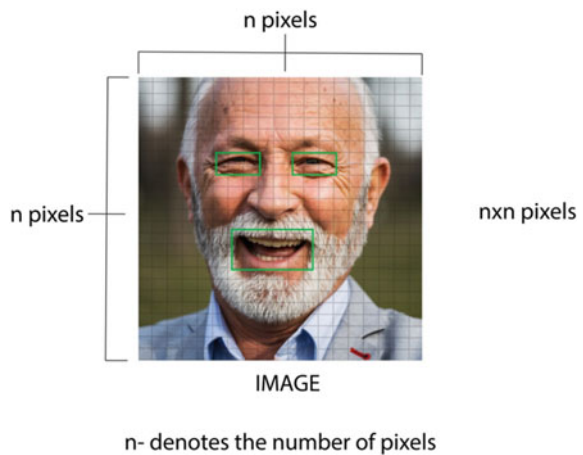




Fig. 4 Emotion recognition

After recognizing the objects present across the viewer. The next step is the processing of the facial expression of the object if the label of the object is classified as person. This part of the proposal is developed for a better analysis of facial expressions for people affected by autism. Emotion recognition may not be a big thing for a normal human but for a person affected by autism who has difficulties in grasping the atmosphere surrounding them. We came up with the solution of emotion recognition, to help autistic persons understand their surroundings and the people they are involved with. YOLOv4 may be helpful in identifying rapid object detection. But it may not be accurate in identifying minute details spread across the image.

In order to achieve greater accuracy and to match the most suitable emotion detected based on the object's facial expression, we use the CNN Model (Fig. 5) to identify facial expressions, by using $N \times N$ pixels and n -layers of neural networks to match the most suitable expression. Here, the sigmoid function helps in transforming the input value into an output value ranging from 0.0 to 1.0 in order for more accurate and minute details. The facial expression recognition (FER) library file with CNN using Tensorflow and Keras library functions is implemented using python for emotion detection.

To Generate and Transfer the Voice Module of the Recognized Object to Ear Aids:

The last process involved in the development of VI-specs (Fig. 6) is the implementation of an earphone that is connected to the VI-specs via Bluetooth signal; here the output is transformed into a voice command to make things convenient for a visually challenged person as well as for an autistic person as well. Here, the google speech API is used for voice recognition and voice commandments.

Voice Module Generator:

Methodology For HE-SPEC (Exclusive specs for a person with a hearing aid):

This proposal is specially meant to design an efficient goggle for a hearing-impaired person. Initially, the device encounters an audio signal which is transferred to the system using Bluetooth, and further, the system does speech to text processing. Finally, the output text will be projected on the glass screen, basically like a sub-title. To achieve this proposal, we need to satisfy the following objectives,

Fig. 5 CNN model

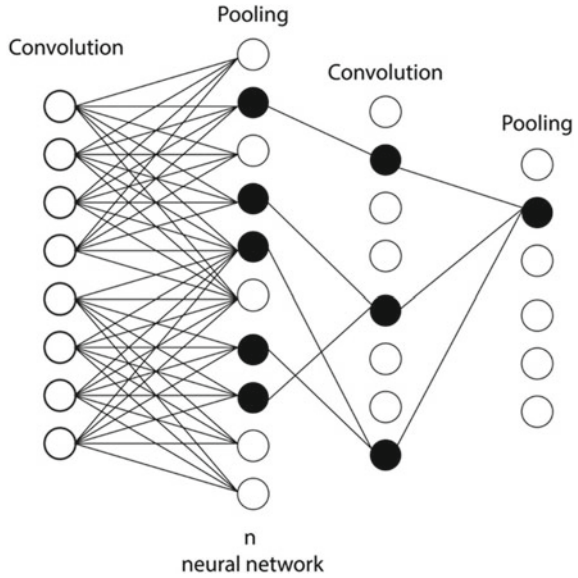
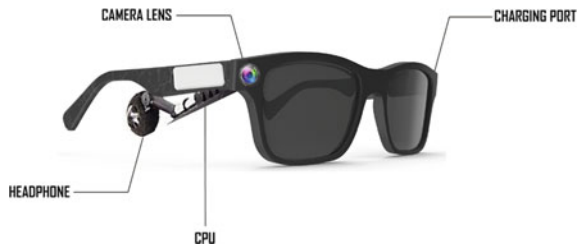


Fig. 6 VI-SPECS design



- To receive the audio input using a microphone.
- To process the audio speech input to text input
- To display the final text information on the screen.

Audio Detection

We all know how an audio signal is captured. This whole process works on the principle of capacitance. The approached audio signal is converted into an electrical signal when it hits the vibrating diaphragm, moving a magnet near the coil.

Speech to Text Processing

This could be done by transferring the received audio clip to the system where it initially identifies the audio segments followed by recognizing the language being spoken. And finally, it converts to a text segment with respect to time code.

Fig. 7 HE-SPEC design

Text Display

The text segments are then sent to the mini projector which displays the final output on the screen of the glass. We have an internal prism to reflect the text segments so that they won't be reflected back on the eyes.

Thus on the whole we have proposed a system that could work efficiently for hearing and visually impaired people. By using this they could sense the function which they lost.

4 Result and Discussion

We live in a world where not every human being is born with all six senses fully operational. People who are born blind or deaf require some assistance in seeing and hearing. We can assist people with autism by using modern technologies such as YOLO 4.

Many technologies intended to facilitate people with autism in seeing well, but none of them convey any information about the emotion of the person in front of them. We developed the VI-SPECS device with the help of YOLO 4 and an expanded camera, which is capable of detecting things and human emotion in real time and communicating it to the appropriate individual.

Hearing aids can help partially deaf persons, but they have zero effect on people who are completely deaf. We use a device called HE-SPECS (Fig. 7), which takes in real-world sound as input and converts it to text messages, which is then displayed. The displayed text is not entirely visible, so it does not obstruct the user's view.

5 Conclusion

YOLOv4 may be an excellent algorithm in rapid and accurate detection of objects but since the VI-specs are made for a broad, yet narrow, view for the user. It may not be helpful in detecting distant objects which may be a problem for a visually challenged person when walking in an open space buzzing with various objects. VI-specs is useful if worn in institutional places such as school, collage, university, etc.

Object detection and classification is made easy with the help of YOLOv4 and OpenCV but describing the features of an approaching object is not conventional in VI-specs.




Processing the object's facial expression using CNN helps in accurate detection of the type of emotion the object is facing but since CNN is a slow process detection compared to other neural network models such as [11] R-CNN, Fast R-CNN, etc. the object's expression may already have changed into something by the time the detection takes place.

References

1. Bochkovsky A, Wang C, Liao HM (2020) YOLOv4: optimal speed and accuracy of object detection. ArXiv, abs/2004.10934
2. Garg D, Goel P, Pandya S, Ganatra A, Kotecha K (2018) A deep learning approach for face detection using YOLO. IEEE Punecon 2018:1–4. <https://doi.org/10.1109/PUNECON.2018.8745376>
3. Xu J, Xia H, Liuand Y, Li Z (2021) Multi-functional smart E-glasses for vision-based indoor navigation. In: 2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM). pp 267–272. <https://doi.org/10.1109/ICARM52023.2021.953608>
4. Arora A, Grover A, Chugh R, et al (2019) Real time multi object detection for blind using single shot multibox detector. Wireless Pers Commun 107:651–661. <https://doi.org/10.1007/s11277-019-06294-1>
5. Go H-J, Kwak K-C, Lee D-J, Chun M-G (2003) Emotion recognition from the facial image and speech signal. In: SICE 2003 Annual Conference (IEEE Cat. No.03TH8734). Vol. 3, pp 2890–2895
6. Bedri A, Li D, Khurana R, Bhuwalka K, Goel M (2020) FitByte: automatic diet monitoring in unconstrained situations using multimodal sensing on eyeglasses. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, pp 1–12. <https://doi.org/10.1145/3313831.3376869>
7. Önal O, Dandil E (2021) Object detection for safe working environments using YOLOv4 deep learning model. Avrupa Bilim ve Teknoloji Dergisi 26:343–351. <https://doi.org/10.31590/ejo sat.951733>
8. Jung T, Kim S, Kim K (2020) DeepVision: deepfakes detection using human eye blinking pattern. IEEE Access 8:83144–83154. <https://doi.org/10.1109/ACCESS.2020.2988660>
9. Chang Y-L, Zeng W, Kamel I, Alonso R (1996) Integrated image and speech analysis for content-based video indexing. In: Proceedings of the Third IEEE international conference on multimedia computing and systems. pp. 306–313. <https://doi.org/10.1109/MMCS.1996.534992>
10. Rao AS, Gubbi J, Palaniswami M, Wong E (2016) A vision-based system to detect potholes and uneven surfaces for assisting blind people. IEEE Int Conf Commun (ICC) 2016:1–6. <https://doi.org/10.1109/ICC.2016.7510832>
11. Long H, Jiang F (2018) IOP Conf. Ser.: Mater. Sci. Eng. 439 032117

Self-balancing Robot Using Arduino and PID Controller



C. N. Savithri , R. S. Roopesh, N Lavanya Devi , P. Shanthakumar, and P. Thirumurugan 

Abstract Robots are programmable machines built to mimic human actions. One such action is locomotion of robots which is the recent area of research. Two-wheeled robots which diligently stabilize it may contribute to the locomotion of robots in upcoming decades. In this paper, the PID controller and Arduino are utilized to design the self-balancing robot. This paper focuses primarily on developing a controller that will aid the robot and test against several parameters such as position, balance along vertical axis and signals for controlling. The accelerometer and gyroscope sensor values are used to determine the precise position of the robot in 3 dimensional space. The sensor values are sent to the controller which controls the rotation of wheels thus aiding in balancing the robot. The two-wheeled robot turns precisely while navigating through different obstacles as against four-wheeled robots.

Keywords Self-balancing robot · Accelerometer · Gyroscope · PID controller

1 Introduction

The advent of computers has led to automation of processes such as controlling the machines automatically. Robots are examples of automated programmable machines capable of performing a series of task. Self-balancing robots are two wheeled robots that can move faster and change the direction with ease. Two wheeled robots are

C. N. Savithri (✉) · R. S. Roopesh
Department of Electronics and Communication Engineering, Sri Sairam Engineering College,
Chennai, Tamil Nadu, India
e-mail: savithri.ece@sairam.edu.in

N. Lavanya Devi · P. Thirumurugan
Department of Electronics and Communication Engineering, PSNA College of Engineering and
Technology, Dindigul, Tamil Nadu, India

P. Shanthakumar
Department of Information Technology, Kings Engineering College, Sriperumpudur, Chennai,
Tamil Nadu, India

basic type that finds applications in autonomous vehicles. Segway and Nine-bot are commercial electric vehicles used as patrol transporters that use the principle of inverted pendulum [1, 2].

In recent years, road accidents have increased due to more vehicles on roads causing property damage and increasing the fatality rate. Every year 1.3 million people pass away due to traffic accidents. Misguided information or reckless driving is responsible for the majority of accidents. According to the survey made, driving with weariness was accountable for 72,000 crashes, 800 fatalities and 44,000 injuries in the United States. Intelligent Transportation Systems (ITS) is gearing up in addition to gaining greater attention owing to their promising approach in improving transportation safety. Wide range of research is done in the area related to self-driving and autonomous cars by many universities and companies.

2 Related Works

A method for detection of lane using Hough transform and detection of drowsiness of the driver is presented. The reasons for poor driving are identified using predictive analytics. The cause for poor driving is also identified in real-time with the help of sight and sound. The paper presents the solution to aid the drivers understand and also to improve their attitude while driving which protects the drivers from being blamed. This helps the driver to maintain the lane and avoid road accidents thereby having an impact on general welfare of people and safety of public [3].

Accurate recognition of objects is performed using radar and vision sensors. The coordinates are different for sensor specific data and hence calibration becomes essential. The algorithms for coordinate calibration between radar and vision images are introduced and calibration of sensors is performed from the data obtained from the sensor [4].

Two wheeled self-balancing robot is designed using PID controller and movement of robot is balanced by solving the problem of inclination angle while implementing in real time. The angle of tilt during load imbalance is measured by fitting the accelerometer on the robot. The results are obtained for stabilization by modeling the robotic system and performing simulation with different control methods [5].

A gyroscope sensor is used to obtain the angular speed of the two-wheeled robot. The angular speed of the motor controlled to maintain the robot balance in upright position is presented. The design of the fractional-order PID controller (FOPID) which has good controllability and the ability to adjust when compared to conventional PID controller is demonstrated. Its performance is validated against PID controller with the help of Matlab simulation. Raspberry Pi using cascaded second-order section form II IIR filters are implemented [6].

A simulation is carried out with level 4 architecture for driving on the highway for the Supplementary Tesla model. Results demonstrate that there is an increase of about 23% in average energy economy against conventional driving. Results

illustrate smoother autonomous control profiles when compared to manual human control which revealed higher levels of fluctuations for velocity profiles and lateral deviations [7].

Two-wheeled self-balancing mobile robot is introduced with control moment gyroscope module that improves the balance while minimizing the movement. A disturbance observer estimates the disturbance and is compensated by the control moment gyroscope, thus maintaining the balance with very less movements [8].

A low cost embedded digital signal processor based driver-assistance system installed in the vehicle is presented. The neighboring lane is considered to be the major cause of accidents. The system used fuzzy based steering and speed regulation to initiate the driving by humans and the developed model was examined in a real-road environment [9].

A framework for providing intelligence to vehicles is introduced. The framework is operated by collecting information like traffic lights, road signs and surrounding vehicles, processes the information intelligently and guides the driver by giving warnings. The obtained simulation results exhibited the merits of the proposed system like simplicity, less response time and scalability achieved by 3G/4G/5G infrastructures [10].

3 Proposed Methodology

The block diagram of a two-wheeled robot, the working principle of the proposed system and the controller action are discussed in this section.

3.1 Block Diagram of the Two-Wheeled Robot

The block diagram of the two-wheeled robot is shown in Fig. 1. The self-balancing robot procures the balance on a pair of wheels by having the grip needed to provide the friction. Two vital points for retaining the vertical axis are measuring the angle of inclination and another is maintaining zero degree with the vertical axis by controlling the forward and backward movement of motors. Accelerometer and gyroscope are the two sensors used for measuring the angle as shown in Fig. 1. The static or dynamic forces are sensed by the accelerometer while the angular velocity is measured by the gyroscope. A math filter known as a complementary filter combines the outputs from the accelerometer and gyroscope. An error value is obtained by subtracting the measured value from the reference value that is already set. The error is applied to the PID controller for control action as shown in Fig. 2.

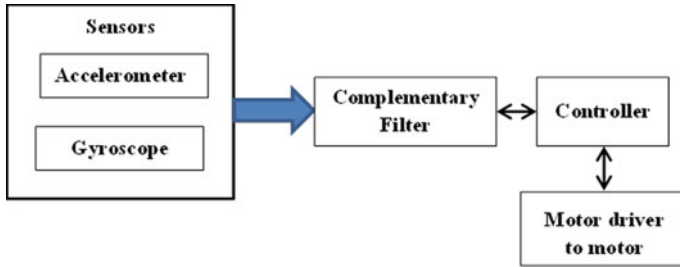
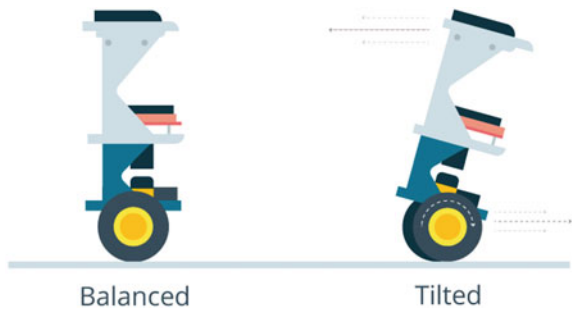


Fig. 1 Block diagram of the two-wheeled robot

Fig. 2 Working Principle of two-wheeled robot



3.2 Working Principle

The two-wheeled robot is able to maintain its balance with the help of the sensors and driving motors only with two wheels. In order to retain balance of the robot, the motors counteract and help the robot from falling by wheels rotating in the desired direction as shown in Fig. 2. This process needs feedback and correcting elements. The acceleration and rotation in all directions is given by the sensors namely gyroscope and accelerometer. The Arduino utilizes this to sense the current orientation of the robot. The motor and wheel constitutes the correcting element of the robot.

The current position of the robot is obtained from the sensors. The orientation of the robot in static condition is given by the accelerometer while the tilt angle is given by the gyroscope. The problem faced while combining the sensor values are.

- (i) The gyroscope returns an incorrect value when it encompasses a drift in a lesser time.
- (ii) The accelerometer returns an incorrect value when it encounters vibrations and returns the correct value while the acceleration is progressive.

A complementary filter is used to overcome the problems occurring due to sensors and it also combines the values from sensors. The complementary filter is a combination of low-pass and high-pass filter. The vibrations from the accelerometer are

filtered out by a low-pass filter while drift from the gyroscope is removed by the high-pass filter. The values thus obtained are combined and applied to the controller for control action. The Eqs. (1) is for the low-pass filter:

$$y(n) = [(1 - \alpha).x(n)] + \alpha.y(n - 1) \quad (1)$$

where $x(n)$ is the value from the accelerometer and $y(n)$ is the output of the low-pass filter.

Equation for high-pass filter is given by Eq. (2)

$$y(n) = [(1 - \alpha).y(n - 1)] + (1 - \alpha)[x(n) - x(n - 1)] \quad (2)$$

where $x(n)$ is the value from the gyroscope and $y(n)$ is the output from the high-pass filter. In Eq. (1) and (2), n represents the current sample and α represents the boundary between values of the accelerometer and gyroscope. The value of α indicates the dependency on current and previous sample values and is assumed to be greater than or equal to 0.5.

The second factor is the angle of inclination (Θ) which is determined by the values of the angles from the sensors as given in Eq. (3)

$$\Theta = [(1 - \alpha).(\text{previous Angle} + \eta_1)] + [\alpha.(\xi_2)] \quad (3)$$

where, η_1 is the angle obtained from the gyroscope, ξ_2 is the angle obtained from the accelerometer.

The angle from the gyroscope (η_1) is obtained by integration and the next value is obtained from the accelerometer (ξ_2). As an example, a zero output from the gyroscope will result in a value given by the accelerometer (ξ_2). A small value of ' α ' discards the output angle from the accelerometer and reads the value from the gyroscope.

3.3 Control Action

The speed of the wheels are controlled by PID (proportional integral derivative), which is a control loop feedback mechanism mainly used in control systems. The calculation of 'error' value is done by the PID. An error value is obtained by subtracting the measured value from the reference value that is already set. The constants of the controller namely K_p , K_I and the K_D are obtained empirically. The error is applied to the PID controller for control action in three steps as shown in Fig. 3.

- (i) A control signal μ is generated by the PID controller when the error is applied as input.
- (ii) The control signal is applied to the two wheeled robot for control action.

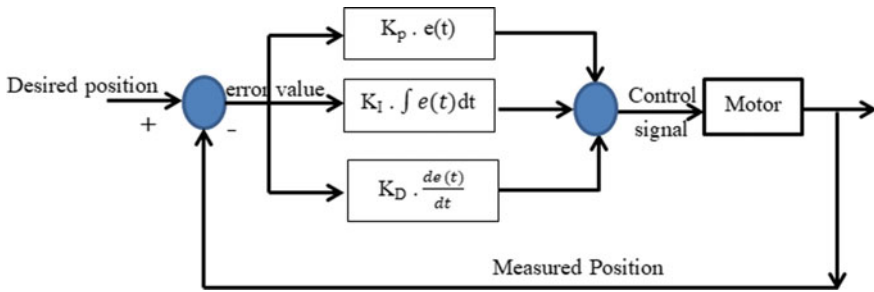


Fig. 3 Flow diagram of the PID controller

- (iii) The two wheeled robot returns back to the reference value in the vertical axis on application of the PID control signal thus gaining the balance.

The microcontroller generates output that drives the motors making the robot stand upright. The response of the PID controller is multiplied by their corresponding constants (K_p , K_d and K_i) and the results are summed. The following steps are used to obtain the values of these constants.

Step 1: The value of K_i and K_d are set to zero and the value of K_p is increased gradually in order for the robot to oscillate about the zero position.

Step 2: The value of K_i is increased to balance the robot when the response of the robot is high. Large K_i is required to make the angle of inclination zero. This brings the robot position back to zero when it is inclined.

Step 3: The value of K_d is increased to reduce the oscillations and overshoots.

Step 4: The above three steps are repeated for tuning each parameter to achieve the desired position.

4 Results and Discussion

The implementation of a two wheeled self-balancing robot is shown in Fig. 4. Figure 4a shows the balancing of robot in the rough surface while Fig. 4b shows the movement of the robot in the smooth surface. The robot is a static linear model and a maximum tilt angle of 12° is observed. The robot moves up to 1 m in the forward as well as backward direction by balancing itself on the flat surfaces. The self-balancing robot balances better on slightly rough surfaces as against the smooth flat surfaces. This is due to the fact that the damping effect of the surface and the moment of inertia required by the wheels is small in contrast to the body of the robot.

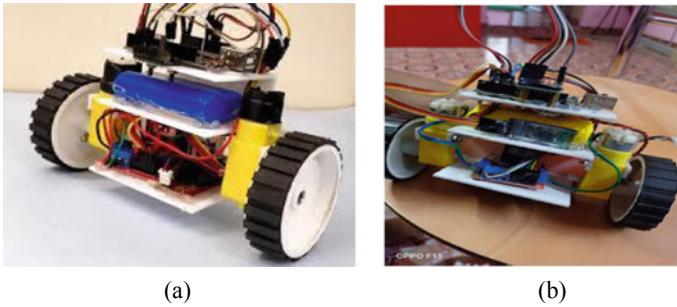


Fig. 4 Set up of two wheeled self-balancing robot **a** Rough surface **b** Smooth surface

5 Conclusion

In this work, a two wheeled self-balancing robot is implemented by Arduino and PID tuning method. The balance of the robot in forward or backward directions is achieved with help of the PID controller. MATLAB are used for upright (vertical) balance angles, their control signals and their disturbance rejection capabilities. A math complementary filter is utilized to remove the drift and vibration occurring in the sensors. Since PID controllers are easier to implement they are used to balance the two wheeled robot. The tilt error is observed to be 5° and the robot is able to handle a load around 250 gms. The maximum tilt angle observed is 12° and is capable of moving 1 m in forward and backward directions. The torque in the DC motor limits the height of the robot which in turn reduces the angular rate. The other limitation is the terrain in which robot can move. For robot to move on slant surfaces, artificial intelligence is required to calculate the slant angle and balance the robot.

References

1. Segway. <http://www.segway.com/>
2. Nguyen HG, Morrell J, Mullens KD, et al (2004) Segway robotic mobility platform. In: Proceedings society of photo-optical instrumentation engineers (SPIE), Mobile Robots XVII, pp 207–220
3. Katyal Y, Alur S, Dwivedi S (2014) Safe driving by detecting lane discipline and driver drowsiness. In: Published in 2014 IEEE international conference on advanced communications, control and computing technologies
4. kim J, Hon D, Senouci B (2018) Radar and vision sensor fusion for object detection in autonomous vehicle surroundings. In: Published in 2018 tenth international conference on ubiquitous and future networks (ICUFN).

5. Chowdhary P, Gupta V, Gupta D, Jadhav A, Mishra V (2018) Design of two wheel self balancing robot using PID controller. *Int J Eng Res Technol* 5(1):1–3
6. Kankhunthodl K, Kongratana V, Numsomran A, Tipsuwanporn V (2019) Self-balancing robot control using fractional-order PID controller. In: *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2019*. IMECS, pp 77–82
7. Divakarla KP, Emadi A, Razavi S (2019) Architecture for autonomous-capable electrified vehicle. *IEEE Trans Transport Electric* 5(1)
8. Park JH, Cho BK (2018) Development of a self-balancing robot with a control moment gyroscope. *Int J Adv Rob Syst* 15(2):1729881418770865
9. Chen Y-L, Wu B-F, Lee TT (2014) Embedded driver assistance system using multiple sensors for safe overtaking manoeuvre. *IEEE Syst J* 8(3)
10. Zographos T, Dimitrakopoulos G, Anagnostopoulos D (2016) Driver assistance through an autonomous safety management framework. In: *Published in 2016 IEEE 12th international conference on wireless and mobile computing, networking and communications (WiMob)*

A Survey Based on Online Voting System Using Blockchain Technology



S. P. Shanthinii, M. Usha, and P. Prittopaul 

Abstract India could be an equitable country which suggests individuals have the control to choose their pioneers. For choice there's a decision handle which is inclined to extortion and has numerous drawbacks. India is losing the genuine meaning of Democracy as the percentage of voting is diminishing definitely day by day. In order to overcome this problem there's ought to give a simple and secured preparation by developing a Mobile/Web Application. This must be considered as each people group's vote plays a noteworthy part in choosing the correct pioneers. Block chain innovation offers the straightforwardness and security requisites for the fair-minded race. It may be a total decentralized, permanent record framework. The online voting framework permits the voters to cast their vote from any place at any time which leads to expanding the voter support check. The objective of the survey is to review a voting framework which gives straightforwardness and security utilizing block chain innovation. The Blockchain framework will permit the citizens to cast their votes in spite of their area. Each year an expansive bunch of Indians closes up the country due to which they are unfit to sharpen their voting rights most of the time. Blockchain system will be a medium for them casting their votes undoubtedly in showing disdain toward the reality that they are not physically displayed in India among race time.

Keywords Distributed ledger technology · Pioneers · Innovation · Online voting · Democracy

1 Introduction

One of the foremost challenging cryptographic convention issues is electronic voting [1]. Bitcoin is the primary decentralized crypto-currency that's right now by far the foremost well-known one in use. The bitcoin exchange language structure is expressive enough to set up advanced contracts whose finance exchange can be upheld consequently [2, 3]. This survey is pointed to a distributed e-voting framework.

S. P. Shanthinii (✉) · M. Usha · P. Prittopaul
Department of CSE, Velammal Engineering College, Chennai, India

Midpoint thought is to incorporate distributed ledger development with a mystery distribution scheme and advanced encryption standard harmonious to realize the distributed e-voting utilization without a trusted mediator. The thought gives an open as well as straightforward vote preparation while ensuring every namelessness of vote caster personality, security about information communication as well unquestionable status of selection amid every charging stage [4]. The Board of Europe is the universal management toward issuing suggestions about every control of utilizing electronic voting [5]. Blockchain was initially conceptualized by Satoshi Nakamoto in 2008 as a center component to bolster exchanges of the computerized cash—Bitcoin, blockchain has been known to be the open record for all exchanges and settled the double-spend issue by combining peer-to-peer innovation with public-key cryptography [6, 7]. This study to begin with clarifies how blockchain makes this enchantment happen and after that presents the blockchain's effective applications in E-voting, E-governance, and E-democracy [8]. There are different strategies including the access to computerized voting all over the world. All abide those associated with diverse prerequisites as well as controversy. One of the foremost imperative along with predominant issues endure the need about inspecting the potential along with framework confirmation strategies. Blockchain innovation, as of late, has picked up a part of consideration that can give an arrangement toward the indicated problem. The stated survey presents Blockchain Voting Framework that portrays electronic voting forms and innards of a direct web voting framework that reviews along with confirmation competent. Blockchain accomplishes this through utilization of blockchain innovation together with voter caster documented paper review path [9]. This survey displays a proposition for a voting system based on blockchain innovation and analyzes its potential to progress current voting frameworks as well as the execution disadvantages [10, 11] tended to the features of blockchain innovation as dispersed record empowers no single point of failure and unused exchange can be embedded into the ledger by anybody who is having the dispersed control.

2 Background

2.1 Overview of Computerized Voting System

By improvement made from Web innovation along with advanced cryptanalysis automation, computerized voting has gotten to be an unused voting strategy, which fathoms many shortcomings of conventional voting strategies, such as tall fetched, moo effectiveness along with frequent botches. Since the primary electronic voting convention was proposed, after more than 30 years of advancement, electronic voting plans based on different cryptosystems have been proposed by numerous cryptologists. Its objective is to provide a secure, helpful and effective voting environment for the Web. These mature arrangements have too been connected to a few government races, corporate board voting and vital decision-making voting.

By and large talking, a protected computerized voting plot ought to contest consecutive necessity [11]:

1. Ballot protection: nobody realizes for the ballot select, along with the sub stance of the ballot is covered up out of possession of the viewer.
2. Individual unquestionable status: Residents can confirm if the votes are checked accurately later balloting.
3. Capability: As it were true blue voters can take an interest in balloting activities.
4. Equity: Nobody could influence the balloting result. It isn't permitted to reveal balloting results or to add ballots within the balloting handle. Something else, it might possess an unjustifiable effect on the balloting result.
5. Singularity: All true blue voters can vote as they once did.
6. Vigor: Nobody's calculation could influence or alter the ultimate balloting result.
7. Keeness: Every poll ought to do numbered accurately.

2.2 Blockchain Technology

The blockchain innovation understands the double-spend issue with the assistance of public-key cryptanalysis, by which each applicant abides doled out a secure vital and a wide vital is shared with all clients. All understanding of blockchain may be a conveyed directory comprising the history of transactions that are mutual among taking section parties. Each of these transactions is proved by the accord of a lion's share of the members within the plan, building false exchanges incapable to pass collective confirmation. Once a history is made and accepted by the blockchain, it could never be changed. Actual inquiries on blockchain have been primarily intent on plan expertise, secure and inventive operations [8].

Through plan, persuasiveness is the foremost critical concern for blockchain. Blockchain must have really strict confirmation preparation to form an unused alternate plan, which leads to a critical idleness of assertion time and squander of computing resources. As of now, it takes almost 15 min for a transaction to be authorized. In expansion, thousands of hubs are running to compute and confirm exchanges [8]. As shown in Fig. 1, blockchain provides transparency, security, smart contract etc.

2.3 Analysis of Ethereum

The survey centers on the sorts of accounts accessible, the transaction structure and the Blockchain convention were utilized in Ethereum. Ethereum has two account types:

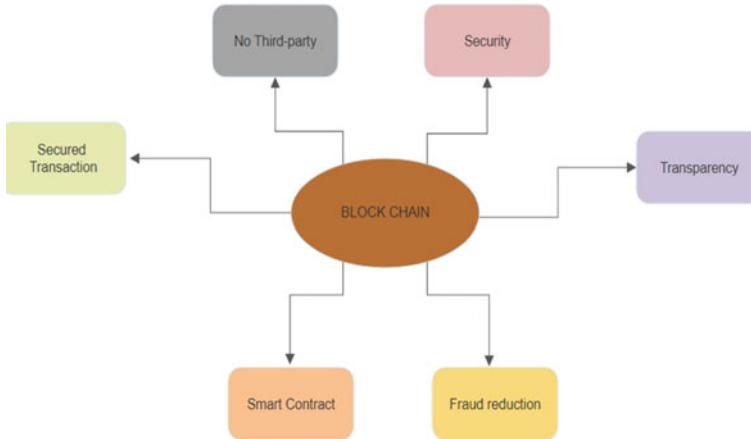


Fig. 1 Structure of blockchain

- A remotely possessed account (user-controlled) could be a public–private key pair controlled by the client.
- The method signifies these accounts by A, B,—A contract account may be a savvy contract that’s controlled by its code. The denotation of a shrewd contract account by λ [12].

The Ethereum-based blockchain is treated as an efficient exchange-based auction model. In the event that different exchanges have a twin obligation, at that point the obligation’s eventual case is decided through arrangement of exchanges that commit in the section. Entirely, Ethereum-based blockchain could be a deviation of the Phantom convention [13] which may be a tree-based blockchain. The indicated tree features a fundamental department in regard to squares that represent the ‘Blockchain’ and exchanges in these squares decide the eventual case of transaction as well as report equilibrium. Comparable to Bitcoin, the guarantee of the Blockchain relies upon mineworkers giving a ‘proof of work’ which approves the miner to add a modern square. The indicated verification effort may be a computationally ambitious amaze, and the digger is compensated 5 ether on off chance that the piece is effectively added.

3 Comparative Study of Blockchain-Based Electronic Voting Plans

See Table 1.

Table 1 Comparative study of blockchain-based voting systems

Theory	Encryption mode	Voting sort	Description	Blockchain techniques
Hardwick et al. [14]	Blink signature	Any	Ethereum blockchain is designed for a deliberate exchange-based auction model	Ethereum
Wu [13], Iversen [15]	Ring signature	1-out-of-m	The conspired blockchain can be partitioned into 3 parts: Creating a key match, producing a ring signature and confirming the signature	Bitcoin
McCrorry et al. [12]	2Round-zero knowledge proof	1-out-of-2	Ethereum blockchain is designed for a deliberate exchange-based auction model	Ethereum and smart contract
Zhao et al. [2]	Zero knowledge proof	1-out-of-m	Each member persuades others that he takes after the convention utilizing zero-knowledge proofs	Bitcoin
Almeida et al. [16]	Vote chain proposal	any	Vote authenticate, casting and counting	Cryptocurrencies
Fujioka et al. [4]	Bit-commitment scheme, digital signature and blind signature	Not specified	This scheme generates a secret voting between the voter and administrator	Communication channel
Hsiao et al. [17], Zyskind [18]	A decentralized electronic identification infrastructure (DI eID), decentralized infrastructure for voting, counting the results (DI voting)	Not specified	DIeID-This foundation ought to gives a method for the solid recognizable proof of users and a list of true blue voters' foundation	Decentralized system

(continued)

Table 1 (continued)

Theory	Encryption mode	Voting sort	Description	Blockchain techniques
Khan et al. [19]	Not specified	Not specified	Bitcoin multi chain generates a contrast for management to design as well as express blockchain function along speed	Bitcoin Multichain
Gao et al. [20]	Certificate less traceable ring signature, code-based, ECC	Not specified		Bitcoin
Chillotti et al. [21], Iversen [15]	Non-malleable encryption, LWE-based homomorphic encryption	Not specified	Uses a voter signature	Cryptography

4 Discussion

A survey is made on online voting systems based on Block chain technology. In Table 1, the comparative study of blockchain-based voting system is made. In these cases, blockchain advances have been conveyed as secure data administration and provenance framework, verification and approval foundation, budgetary settlement foundation, and exchange administration foundation. To encourage, create and develop blockchain activities, the UK Government Chief Logical Office has given a few suggestions in progressing blockchain advancements in the government and society, which incorporate [22]: (1) setting up a ecclesiastical level authority to guarantee that government gives the vision, authority and the stage for conveyed record innovation inside the government; (2) that the investigation community contributes within the inquiry about that required to guarantee if disseminated records are adaptable, secure and give confirmation of rightness of their substance; (3) that government bolsters the creation of conveyed record demonstrators for nearby government that solidifies all the components fundamental to test the innovation and its applications; (4) government ought to put in put the fundamental administrative system for dispersed record. This study examined all parameters for voting framework counting security. In spite of the fact that security is guaranteed, future work can upgrade the security indeed.

5 Conclusion and Future Work

The survey recognized the set of center standards for equitable balloting along with the most discussed settings of blockchain-based electrical balloting systems. The survey diagram depicts the advancement handle of electronic voting. The survey considers that blockchain innovation could be utilized to establish electronic voting plans to accomplish nearly every standard for law-based elections. However, there is caution against analysts putting away ballots over open blockchains compensation to long-haul protection commerce. Electronic voting system is conveyed on Ethereum arrange. Numerous research works demonstrated that the square chain innovation makes a difference in improving the existing framework; subsequently it gives a better way to conduct the Race. It is too utilized to avoid the drawbacks of centralized voting frameworks. This survey concludes that an electronic voting system can be designed using Ethereum with more security and transparency.

In future, the security can be enhanced even better by including an Iris sensor as well as a fingerprint sensor to check whether the user is a valid user or not. Future work could implement additional security.

References

1. Kiayias A, Yung M (2002) Self-talking elections and perfect ballot secrecy. In: International workshop on public key cryptography. Springer, pp 141–158
2. Zhao Z, Chan TH (2015) How to vote privately using bitcoin. Springer
3. Bentov I, Kumaresan R (2014) How to use bitcoin to design fair protocols. In: Garay JA, Gennaro R (eds) CRYPTO 2014, Part II, vol 8617. LNCS. Springer, Heidelberg, pp 421–439
4. Fujioka A, Okamoto T, Ohta K (2019) A practical secret voting scheme for large scale elections. In: Proceedings of the international workshop on the theory and application of cryptographic techniques. Springer, pp 244–251
5. Bernhard M, et al (2017) Public evidence from secret ballots. In: Krimmer R, Volkamer M, Braun Binder N, Kersting N, Pereira O, Schürmann C (eds) e-VoteID 2017. LNCS, vol 10615. Springer, Cham, pp 84–109
6. Ruffing T, Moreno-Sanchez P, Kate A (2014) CoinShuffle: practical decentralized coin mixing for bitcoin. In: Kutyłowski M, Vaidya J (eds) ESORICS 2014. LNCS, vol 8713. Springer, Cham, pp 345–364
7. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system
8. Zhao JL, Fan S, Yan J (2016) Overview of business innovations and research opportunities in blockchain and introduction to the special issue. In: Financial Innovation. Springer, Berlin Heidelberg, pp 2–28
9. Ojo A, Adebayo S (2017) Blockchain as a next generation government information infrastructure: a review of initiatives in D5 countries. In: Government 3.0—next generation government technology infrastructure and services. Springer, Cham, pp 283–298
10. Pawlak M, Guziur J, Poniszewska-Marañda A (2018) Voting process with blockchain technology: auditable blockchain voting system. In: Xhafa F, Barolli L, Micha Á, Gregu Á (eds) Advances in intelligent networking and collaborative systems, INDECT 23, ISBN 978–3–319–98556–5. Springer-Verlag, Heidelberg
11. Xiao S, Wang X Information security of the PAP. Engineering University of the PAP, Xi'an Shaanxi, China, p 710086

12. McCorry P, Shahandashti SF, Hao F (2017) A smart contract for boardroom voting with maximum voter privacy. In: International conference on financial cryptography and data security. Springer, pp 357–375
13. Wu Y (2017) An e-voting system based on blockchain and ring signature. University of Birmingham, Master
14. Hardwick FS, Akram RN, Markantonakis K (2018) E-voting with blockchain: an e-voting protocol with decentralisation and voter privacy. CoRR abs/1805.10258 (2018). arXiv: 1805.10258
15. Iversen KR (1992) A cryptographic scheme for computerized general elections. In Advances in cryptology—CRYPTO '91. Lecture Notes in Computer Science 576. Springer-Verlag, Berlin, pp 405–419
16. Almeida RL, Ricci L, Camarinha-Matos LM (2019) Votechain: community based scalable internet voting framework. In: Doctoral conference on computing electrical and industrial systems. Springer, pp 70–80
17. Hsiao J-H, Tso R, Chen C-M, Wu M-E (2017) Decentralized e-voting systems based on blockchain technology. In: Advances in computer science and ubiquitous computing. Springer, pp 305–309
18. Zyskind G, et al (2015) Decentralizing privacy: using blockchain to protect personal data. In: Security and privacy workshops (SPW). IEEE, pp 180–184
19. Khan KM, Arshad J, Khan MM (2020) Investigating performance constraints for blockchain based on secure e-voting systems. Future Gener. Comput. Syst. 105:13–26
20. Gao S, Zheng D, Guo R, Jing C, Hu C (2019) An anti-quantum E-voting protocol in blockchain with audit function. IEEE Access 7:115304–115316
21. Chillotti I, Gama N, Georgieva M, Izabachène M (2016) A homomorphic lwe based e-voting scheme. In: Proceedings of the 7th Post-quantum cryptography. Springer, pp 245–265
22. Taylor S (2016) Distributed ledger technology : beyond block chain

Survey on Collaborative Filtering Technique for Recommender System Using Deep Learning



S. L. Jothilakshmi  and R. Bharathi 

Abstract Automated systems like Recommenders are developed to recommend item suggestions to consumers abiding a variety of conditions. They predict ratings of the most likely product a user will buy intuitively or based on interest. Collaborative filtering (CF) is one of the most common methods for designing recommender systems. It works on a collection of users' established preferences to make suggestions or predictions for unknown users. This survey study provides a comprehensive insight into the most up-to-date state-of-the-art approaches for efficient filtering and successful recommendation. In addition, the survey article studies different types of deep learning-based collaborative filtering algorithms and implementation techniques used in different recommender system applications to overcome cold start and data sparsity challenges.

Keywords Deep neural networks (DNNs) · Collaborative filtering approach · Recommender systems (RS)

1 Introduction

Deep learning is a relatively new area of research in machine learning. It imitates the human brain's process of interpreting data such as pictures, sounds, and texts. The deep learning approach is divided into supervised and unsupervised learning, much like machine learning. To discover distributed feature representations of data, deep learning combines low-level features to shape more abstract high-level representation attribute categories or features [1]. Deep learning is often referred to as "new-generation neural networks" because it was first coined in the area of artificial neural networks. There has been a remarkable achievement in many application fields such

S. L. Jothilakshmi (✉)

Amrita College of Engineering and Technology, Nagercoil, Tamil Nadu, India

e-mail: sl_jothilakshmi@amrita.edu.in

R. Bharathi

University College of Engineering, Nagercoil, Tamil Nadu, India

as machine visions, Natural Language Processing, Audio recognition, and Recommender Systems (RSs), since the advent of the deep neural network. Researchers have been working exclusively to expand the use of deep neural networks in a variety of applications [2].

Deep learning offers a deeper understanding of user needs, item characteristics, and historical experiences between them than conventional recommendation models. Deep learning for recommender systems has sparked a lot of interest, and several deep learning-based recommendation models have been proposed [3]. They could be used for movie recommendations, content-based image recommendations, website recommendations, and Document Context-Aware Rating Prediction [1–3]. In recommender systems, the user and item ratings are combined to predict the outcome. For example, MudRecS, makes suggestions based on the ratings of multimedia objects that fit a user’s interests [4]. Deep learning-based Collaborative filtering is one of the most widely used strategies in recommender systems [5].

The novelty of this survey article is that we investigated the use of several deep collaborative filtering (CF) models to evaluate the relationship or association between users and the items in interest. Traditional CF-based methods, such as the restricted Boltzmann computer, can only understand a single form of relationship, such as the user–user or item–item relationship [6].

E-commerce platforms produce a large amount of data on today’s industry. As a result, Recommender Systems (RSs) are a technique to deal with information overload and boost consumer satisfaction by providing personalized suggestions [7]. Many application domains, such as e-commerce, movie and music reviews, tourism, news, marketing, financial markets, and social networks employ collaborative filtering and deep learning approaches to enhance the efficiency of recommendation tasks [8].

This paper provides a detailed analysis of deep learning combined with a collaborative filtering approach for recommender system applications. We discuss the following three relevant concepts: (i) The impact of deep learning technology in recommender systems. (ii) The features of the collaborative filtering approach in recommender system applications. (iii) Different deep learning techniques. The remaining part of this paper is structured as follows: Section 2 presents the detailed view of deep learning in recommender systems. Section 3 shows various applications that use deep learning-based collaborative filtering approaches.

2 Recommender Systems

Recommender systems are used to accurately identifying users’ preferences and choosing suitable items from a large volume of data. The recommendation algorithm is at the heart of the recommender system. To perform recommendation tasks, the current recommendation systems use a variety of strategies and models for learning user preferences [9]. They are categorized as follows:

- **Content-based recommendation systems:** Here recommendations are based on the content of products which the consumer has previously shown interest in. e.g. Genre of the movie.
- **Collaborative Filtering (CF) based recommendation systems:** Here recommendations are based on previous similar preferences.
- **Hybrid recommendation systems:** Recommendations are based on content-based and Collaborative Filtering based approaches.

The content-based model has limited ability to expand on the users existing interests only. To avoid these limitations collaborative filtering techniques can be used. Collaborative filtering techniques are divided as shown [10].

1. **Memory-based algorithms:** Memory-based algorithms are heuristic algorithms estimating a target user's rating for an object by using partial knowledge of the target user as well as adjusted weights generated from the dataset.
2. **Model-based algorithms:** Model-based algorithms are machine learning approaches used for detecting patterns in CF datasets. Matrix factorization, is a potential candidate in capturing the interaction between user and item ratings directly.

Recommendation systems face the following issues:

- **Cold start:** There is insufficient information available when a new person or product is added to the recommendation system for the first time. This is known as the "cold start" problem. Because fewer customers have tried a new item, it takes a long time to incorporate it into the recommendation process [11].
- **Data Sparsity:** The condition is known as data sparsity when consumer input is inadequate for recommendation system models to operate on. It may be due to a cold start or the users' mistakes for not providing enough input. This has the potential to affect the selection and consistency of recommendations [12].
- **Scalability:** When the number of users and objects grows, so does the cost of computation. This may affect the recommendation system's efficiency and response time.

2.1 Deep Learning-Based Recommendation Systems

Algorithms that recommend relevant items to users are called recommender systems. In recommender systems, the recommendations are mainly based on implicit interactions like clicks, listened to songs, watched movies, likes, dislikes of users with item ratings. The dataset comprises both implicit and explicit feedback. It is hard to explain the distinction between implicit and explicit input in recommender systems, as well as why existing recommendation systems depend on implicit feedback. Generally, there are two phases involved in the recommendation system. The first phase discusses feature extraction using deep learning techniques. The extracted features are the



Fig. 1 Architecture of deep learning based recommender system

input for the next phase. The second phase consists of a deep collaborative filtering approach for recommendations. Figure 1 shows architecture of Deep learning based recommender system.

2.2 Deep Collaborative Filtering Techniques

The key challenges in collaborative filtering algorithms for recommendation systems are to create effective suggestions overcoming data sparsity and cold start. For this issue, several ways are proposed. The majority of them train the models using one source of data, while other methods perform poorly, especially while data sparsity is high. DeepHCF a deep learning-based model employing joint training to train two deep models by exploiting rating matrix and item reviews, which is potential in addressing this issue. The user-item rating matrix data is learned using the Multi-Layer Perceptron (MLP), whereas the other side information is learned by Convolutional Neural Network [13].

Deep learning-based collaborative filtering methods can outperform traditional methods in terms of recommender system performance, especially when text data is available. The proposition is the proposed network constitutes more layers to extract information embedded in the product's text description and combines with the rating information. As the network grows in-depth, preamble signal attenuation may cause failure in convolutional neural network. Hence the gradient progressively fades away altogether in the back-propagation process thus preventing weights from being altered. A new approach, eXtreme Residual connected Convolution Collaborative Filtering (xRConvCF) is plausible in estimating the rating for each item considering the textual input [14].

The majority of deep collaborative filtering-based recommender systems uses Convolutional Neural Networks, Recurrent Neural Networks, Multilayer Neural Networks (MNN), and auto-encoders to provide consumers with suggestion ratings [13, 14]. Deep Factorize-based Machines (deep FM) and Neural Collaborative Filtering (NCF) used in CF neural approaches. Using a dual neural network, NCF processes item and user information at the same time [2, 15, 16]. For complex recommender tasks, deep learning-based techniques such as deep neural networks and convolutional neural networks manage a vast amount of data [17]. The hybrid model-based technique was implemented using a combination of generative, discriminative, and deep learning models.

2.3 Datasets

There are many benchmark datasets available for implementing the deep collaborative filtering-based recommender applications. Some of the benchmark datasets available in the literature are MovieLens (Movie Recommendations) [4], Pinterest (Top-N Recommendation) [6], TripAdvisor [5], amazon dataset [14]. The performance of various approaches is evaluated with the help of benchmark datasets using the following metrics: root mean square error (RMSE) [1, 3, 7] and hit ratio (HR) [2].

3 Recommendation System Applications

This section summarizes some recommendation applications such as movie recommendation, hotel recommendation, and music recommendation. These applications used deep learning methods for achieving successful recommendations. Other applications such as book recommendation, top N recommendation, document recommendation, also utilize the deep collaborative filtering approach for better performance.

Recommendation systems are essential for streaming video services like Netflix in assisting consumers and finding new movies to enjoy. Lund and Ng [4], suggested a deep learning technique that uses auto-encoders to construct a distributed filtering framework that predicts a user's movie ratings based on a large database of other users' ratings. They look at how deep networks may be used to provide movie recommendations in the MovieLens dataset and predict user ratings on new films. The k-nearest-neighbor and matrix-factorization approaches were used to validate the deep learning method's novelty and accuracy.

Gupta et al. [18], developed an item-based method (collective filtering) for the suggestion, which is considerably more accurate and easy to use than the user-based approach because it can be done offline and is not dynamic. The distance between the target and all other movies in the dataset is calculated using the KNN method, and the top K- most closely related movies are ranked using a cosine similarity metric in the filtered products.

Zhang et al. [14], used a deformable convolutional network (DCN) to improve the capability of using an offset layer to the standard convolution layer and work on a new deformable convolutional network matrix factorization (DCNMF) recommendation model that results in a convolution transformation model. The DCNMF model was tested on three real-world datasets MovieLens 1 m, MovieLens 10m1, and Amazon. The performance evaluation used root mean squared error (RMSE) [14].

The tourism industry is one of the most primary applications of recommendation systems. The collaborative filtering technique based on deep learning and the multi-criteria rating was developed by Alfarhood and Cheng [3]. This model extracts user and item ratings as input to the proposed deep neural network's criteria ratings anticipation. The results are evaluated using TripAdvisor1, a multi-criteria rating

dataset containing hotel ratings from users. It comprises of several criteria ratings like Value, Place, Hygiene, Office Desk Check-in, Services, with an overall rating ranging from 1 to 5. The Mean Absolute Error F-measure is used to assess the model's efficiency.

Wang et al. [19], proposed a feed forward neural multi-criteria task collaborative filtering algorithm (NMCF) which maps the possible relationship between the user ratings and confidence relationships to achieve mutual promotion that leads to a better prediction of rating, rather than oblivious in optimizing the training parameters.

Table 1 summarizes the widely used recommendation applications with their state-of-the-art approaches. Deep learning-based Recommender models are better representations of learning the user-item experiences compared to conventional recommendation architectures. Similarly, in some implementations, deep learning can be used together with collaborative filtering to achieve good results. To conclude, the main limitation of recommendation system using the collaborative filtering approach is the phenomenon of the cold-start problem due to non-availability of information about an item or a user to make relevant predictions [18]. Generally, to avoid the cold start problem a matrix factorization—a neural collaborative filtering approach—is used in a personalized recommendation system [20]. DeepHCF is used on deep learning recommendation model using multi-layer perceptron in the convolutional neural network thus creating a hybrid collaborative filtering approach capable of tackling two separate sources of input data [13]. In the future, a hybrid-model approach using an online dataset will be developed for improving the accuracy of the recommender system.

4 Conclusion

A systematic overview of deep learning-based recommender systems is shown in this article. Deep learning-based recommender systems could learn user and item ratings from large amounts of data, and then build a recommendation model using a deep collaborative filtering technique. First a detailed discussion about deep learning approaches in recommender systems is shown, along with datasets for Deep Collaborative Filtering in Recommender Systems. Second a quick outline of the various deep learning and collaborative filtering applications was provided. The development of efficient deep neural networks, a robust recommender framework, and benchmark datasets for complex online recommendation tasks all are key future directions.

Table 1 Comparison of different existing methods in recommendation system Applications

Recommendation system application	Ref	Relations	Metrics	Deep learning techniques	Dataset	Collaborative filtering techniques
Movie recommendations	[4]	Movie name, individual movie ratings, time stamp	Root mean square error (RMSE)	Auto encoders	MovieLens	k-nearest-neighbor and matrix-factorization
Top-N Recommendation	[6]	Same director same actress same producer	Hit ratio, normalized discounted cumulative gain	DeepICF	MovieLens & Pinterest	Item-based CF
Document context-aware rating prediction	[14]	Plot and story of movies	Root mean square error (RMSE)	CNN stochastic gradient descent (SGD)	MovieLens 1 m (ML-1 m) and MovieLens 10 m (ML10)	Residual connected Convolution Collaborative Filtering (xRCConvCF)
Multi-task-based recommendation systems	[19]	Ratings of movies and friendship information	RMSE and MAE	Deep learning	Film trust Flixster	Neural multi-task collaborative filtering (NMCF)
Content-based image recommendations	[20]	Photographs	hit ratio (HR), normalized discounted cumulative gain	Probabilistic auto-encoders and ridge regression	Pinterest ML-1 M	Neural embedding collaborative filtering

References

1. Mu R (2018) A survey of recommender systems based on deep learning. *IEEE Access* 6:69009–69022. <https://doi.org/10.1109/ACCESS.2018.2880197>
2. Da' u A, Salim N (2020) Recommendation system based on deep learning methods: a systematic review and new directions. *Artif Intell Rev* 53:2709–2748. <https://doi.org/10.1007/s10462-019-09744-1>
3. Nassar N, Jafar A, Rahhal Y (2020) A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowl Based Syst* 187
4. Lund J, Ng Y (2018) Movie recommendations using the deep learning approach. *IEEE Int Conf Inform Reuse Integ (IRI) 2018:47–54*. <https://doi.org/10.1109/IRI.2018.00015>
5. Aljunid MF, Dh M (2020) An efficient deep learning approach for collaborative filtering recommender system. *Proc Comp Sci* 171:829–836. ISSN 1877–0509. <https://doi.org/10.1016/j.procs.2020.04.090>
6. Xue F, He X, Wang X, Xu J, Liu K, Hong R (2019) Deep item-based collaborative filtering for Top-N recommendation. *ACM Trans. Inf. Syst.* 37(3):1–25. <https://doi.org/10.1145/3314578>
7. Alamdari PM, Navimipour NJ, Hosseinzadeh M, Safaei AA, Darwesh A (2020) A systematic study on the recommender systems in the E-commerce. *IEEE Access* 8:115694–115716. <https://doi.org/10.1109/ACCESS.2020.3002803>
8. Kulkarni S, Rodd SF (2020) Context aware recommendation systems: a review of the state of the art techniques. *Comp Sci Rev* 37:100255. ISSN 1574–0137. <https://doi.org/10.1016/j.cosrev.2020.100255>.
9. Mu R (2018) A survey of recommender systems based on deep learning. *IEEE Access* 6:69009–69022. <https://doi.org/10.1109/ACCESS.2018.2880197>
10. Bobadilla J, Alonso S, Hernando A (2020) Deep learning architecture for collaborative filtering recommender systems. *Appl Sci* 10(7):2441
11. Zhao VN, Moh M, Moh T (2016) Contextual-aware hybrid recommender system for mixed cold-start problems in privacy protection. In: 2016 IEEE 2nd international conference on big data security on cloud (Big Data Security), IEEE international conference on high performance and smart computing (HPSC), and IEEE international conference on intelligent data and security (IDS), pp. 400–405. <https://doi.org/10.1109/BigDataSecurity-HPSC-DS.2016.54>
12. Fayyaz Z, Ebrahimian M, Nawara D, Ibrahim A, Kashaf R (2020) Recommendation systems: algorithms, challenges, metrics, and business opportunities. *Appl Sci* 10:7748. <https://doi.org/10.3390/app10217748>
13. Alfarhood M, Cheng J (2018) DeepHCF: a deep learning based hybrid collaborative filtering approach for recommendation systems. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA). pp 89–96. <https://doi.org/10.1109/ICMLA.2018.00021>
14. Zhang B, Zhu M, Yu M, Pu D, Feng G (2020) Extreme residual connected convolution-based collaborative filtering for document context-aware rating prediction. *IEEE Access* 8:53604–53613. <https://doi.org/10.1109/ACCESS.2020.2981088>
15. Fu M, Qu H, Yi Z, Lu L, Liu Y (2019) A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Trans Cybernet* 49(3), 1084–1096. <https://doi.org/10.1109/TCYB.2018.2795041>
16. Unger M, Tuzhilin A, Livne A (2020) Context-aware recommendations based on deep learning frameworks. *ACM Trans Manage Inf Syst* 11(2):1–15. <https://doi.org/10.1145/3386243>
17. Chen H et al (2019) Deformable convolutional matrix factorization for document context-aware recommendation in social networks. *IEEE Access* 7:66347–66357. <https://doi.org/10.1109/ACCESS.2019.2917257>
18. Gupta M, Thakkar A, Gupta V, Rathore DPS (2020) Movie recommender system using collaborative filtering. In: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). pp 415–420. <https://doi.org/10.1109/ICESC48915.2020.9155879>

19. Wang S, Cheng M, Ma Z, et al (2020) Neural multi-task collaborative filtering. *Evol Intel* <https://doi.org/10.1007/s12065-020-00409>
20. Huang T, Zhang D, Bi L (2020) Neural embedding collaborative filtering for recommender systems. *Neural Comput Appl* 32:17043–17057. <https://doi.org/10.1007/s00521-020-04920-9>

A Survey on Power Consumption Indicator Using Machine Learning-Based Approach



R. Hamsini and P. Visu 

Abstract According to the Electricity market, power consumption seems to be the actual energy demand which is made on the existing electricity supply. Recommendation systems play a significant role in Machine Learning and Artificial Intelligence. Recommender systems also plays a crucial role in majority of the industries. It uses various approaches in predicting user's interest and provides relevant recommendations by using different filtering methods. This paper has a detailed survey on the recommender systems based on power consumption. Recommender systems provide different algorithms for an effective way of consuming power. Residential customers need to consume power efficiently and also can enjoy the alert system which reminds them of their monthly usage which inturn helps the customers to manage their electricity bill. In our proposed system the Power Limit Indicator system (PLIS) manages the usage of power for the individual residential customer. Residential customers of the PLIS are alerted based on the data of household appliances when their limit reaches the threshold. The PLIS outperforms other strategies which seem to provide a strong solution to power limit alert task.

Keywords Recommendation systems · Power limit indicator system · Power limit alert

1 Introduction

Power Consumption grew 18.6% due to improved economic activities and has recorded yearly growth of about 4.6% in September 2020. There is a huge power demand and the rate of power consumption is increasing rapidly. Energy demand is increasing rapidly and thus integrating renewable energy sources with non-renewable

R. Hamsini (✉) · P. Visu
Velammal Engineering College, Chennai 600066, India
e-mail: hamsiniraj8@gmail.com

P. Visu
e-mail: visu.p@velammal.edu.in

resources is widely followed. Electricity tariffs have been impacted in recent years. Large power consumption increases the electricity tariff. The residential users have gradually understood the current power consumption scenario. With increase in power consumption, there are various methods to reduce power consumption in commercial and industrial sources.

The energy consumed by the residential user is rapidly increasing. This is due to the non-identification of the electricity consumption for individual residential users. This involves various recommendation systems to provide individual residential tariffs. The recommendation system has a wide range of attention in the past decades. With the wide spread of Internet, given the opportunity to interact with the catalogue of millions of items on different platforms, user can have a great experience when it is integrated with recommendation systems. With these systems, tariffs are suggested to the end users based on their requirement.

The residential electricity tariff suggestion seems to reduce the power consumption and also the users have the flexibility to choose their own plan according to their requirement. Even after choosing the tariff, it seems that the residential users are not aware of the energy consumed by them. Over consumption of power results in the huge metric change in the tariff. Thus, the recommendation systems can be integrated with various filtering methods and provide an alert when the individual residential customer exceeds the limit of their regular usage. This way helps in consuming the power efficiently and also manage their tariffs effectively.

2 System Description

Recommender systems provide an efficient algorithm for power consumption. Power Consumption Alert will help the users to effectively use the power without wasting it. In our proposed system, Power Limit Indicator system (PLIS) alerts and manages the usage of electricity for the individual residential customer.

2.1 *Power Limit Indicator System (PLIS)*

Power Limit Indicator System will alert the individual user on the power consumption. We can have three Phases where the first phase will access the customer data and the second phase can use different filtering algorithms to detect the power consumed and the last phase can alert the user based on the acquired results. This may even reduce the individual user's power consumption. Thus makes a huge difference in the residential power supply.

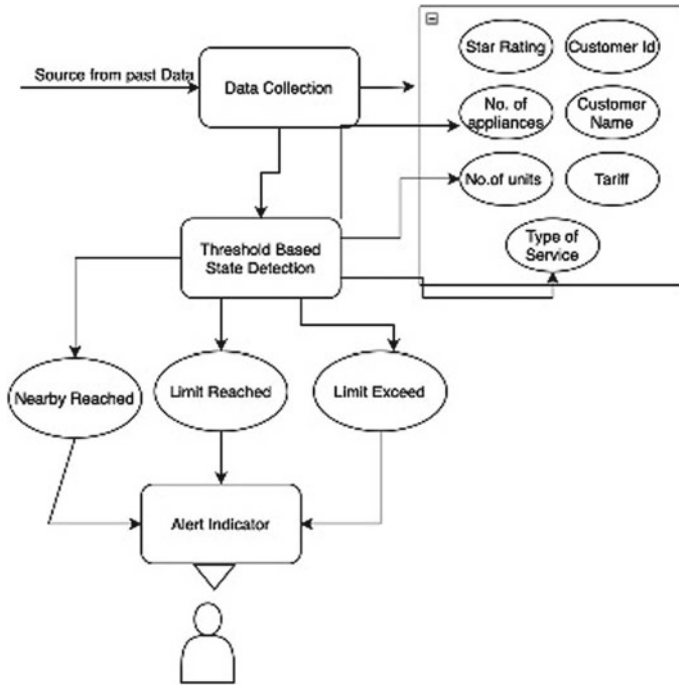


Fig. 1 PLIS flow

2.2 PLIS Basic Flow

As shown in the Fig. 1, the first phase of PLIS accesses all the customer data. Data set with customer ID, type of user (residential/commercial), number of appliances used, star rating of the appliances, customer data usage (past years) are used. Based on the data in the first phase, detection of power consumption takes place in the second phase. Here, based on the recommendation systems and different filtering algorithms, suggestions are made. And also, alert measure is generated in this phase. The final phase provides the alert indicator. It alerts the end user in three forms—nearby reached/reached limit/limit exceed. This helps the individual residential user to plan their energy consumption which in turn reduces the power consumption rate.

3 Review and Analysis of Power Consumption Systems

The proposed system [1] performs the data analysis, and also per-process and trains test split even before training the model. This proposed system trains a stable model that performs well in all aspects compared to the previous model with various metrics

errors like mean absolute, mean absolute percent, mean squared, and root mean squared logarithmic. The results obtained show that the energy from the fossil fuel is very relatively high when compared to the other forms of renewable energy. Recommender system [2, 3] provides efficient guidance to the customers to select their suitable electricity plans and tariffs. This system uses collaborative filtering for higher precision of the recommendation results. It introduces a weighted metric from appliance into Jaccard Euclidean which helps to improve the missing feature. This model also introduces fast nearest neighbor detection and recommendation plan based on the retailing market policy. The proposed system [4, 5] provides an efficient method in saving the energy consumed by targeting the unoccupied spaces and relaxing the set point temperature. This system uses two types of recommendations—moving recommendation and shift schedule recommendations. This system proposes a tightly coupled system comprised of the simulator and recommendation system. The model produces 25% more energy consumption in the simulation results.

This system [6] helps us in understanding the various machine learning techniques in building the energy prediction and analysis. A detailed review on four modelling criteriamachine learning techniques, deep learning models, optimization techniques, and the geospatial distribution to design and analyze the energy consumption. It also provides various computational machine intelligence techniques and relevant architecture used to design energy consumption. The proposed model [7] provides comprehensive update of the major machine learning models. Though there are lot of conventional Machine Learning methods such as ANNs and MLP, DTs, application of hybrids and Ensemble methods, this paper suggests hybrid methods for higher accuracy and speeds. This research reports that using the novel hybrid and ensemble predication models provides an outstanding increase in the performance of the prediction technologies. This model [8] provides an (EM) 3 which is a combination of data collection, abstraction, recommendations. This system handles many sensor data efficiently to provide recommendations. (EM) 3 currently works with recommendations based on rules and the results of the predictions tend to have increased energy efficiency and scalability. Future prototype of (EM) 3 will extend it to more users and also to improve user energy profiles. The proposed system [9] proposes a task scheduling which is said to be optimal for different electrical constraints which also ensures the comfortability of the thermal according to the user needs. The proposed algorithm helps in identifying the optimized solution to the imposed constraints. It also allows the user to have relevant flexibility in choosing huge working horizons and lower the resolution values. The overall optimization framework from a thermal view will be the future efforts.

The proposed model [10] provides recommendations on context-aware adaptiveness for its users for energy savings. To increase effectiveness and produce high manageable change of behaviors, it relies on the gamification elements. This model prevents high dropout rates, provides an in depth understanding of the deployments of various aspects like original mix of instruments, gamification-based stimuli, web and mobile user interfaces, and recommendation for personalized energy saving for sustainable energy consumption. This model [11] proposes an expert-based approach which prioritizes the demand curtailment allocation using the AHP method. Various

Demand curtailment parameters such as loading level, capacity of demand response, customer type, and the loading categories are used in the Peak Demand Management system. The simulation results from the proposed system tends to achieve Peak load reduction. This system [12] works under scheduling of hourly costing and limiting the peak power. The Scheduling system optimizes the cost reduction in a dynamic pricing environment. It works on the demand response strategies which include all controllable assets which are thermostatically controllable and non-thermostatically controllable. There are different parameters like the energy storage system and distributed generation that are considered in this scheduling system. This system uses several load shaping straggles to minimize the peak power. The goal of the system is to optimize the total cost of the household. The simulation results from the proposed system seems to be computationally efficient. This system [13] provides optimization techniques to minimize the conventional energy and also diminishes the cost of consumption. This paper proposed a game theoretic consumption which uses a mixed integer programming. This system schedules a consumption plan for residential customers. The simulation results from the proposed system could detect a stable solution and also admits a Nash equilibrium. It proves that the cost of energy is minimized when renewable energy is integrated with the consumption scheduling optimization mechanism.

The proposed system [14] provides a detailed study on the recommendation systems. This study mainly concentrated on context-aware which is a type of recommendation system for efficient energy consumption. It extends the persuasive recommendations which are totally based on user preferences. The objective of the recommendation system in power saving action is based on either one of the persuasive facts such as economical or on an ecological saving prospect or impact. The obtained results yield 19% increase in economical or ecological facts. The current results are based on the recommendation acceptance ratio parameter. These systems [15] describe the properties and challenges of the recommendation systems in locating services. It provides a comprehensive survey on the objective, methodology and the data source for the recommendation systems. The objective of the recommendation includes various parameters like the locations, activities, users or the social media. The methodologies of the recommendation systems include Link analysis-based methodology, content-based methodology, collaborative filtering-based methodology. The data sources of the recommendation systems include user history online, user history location and user profiles. This paper helps in researching the location-based recommendation systems with a balanced depth and scope. The proposed model [16, 17] describes the properties and challenges in the social media recommendations. This paper address relatively huge problems like social related, recruitment of the members, and recommendations for friends. The major challenge will be the heterogeneous data structures. Session based recommendations, cross domain recommendation, social trust ensemble learning model are some of the listed deep learning techniques. This model [18, 19] provides a detailed review on the energy saving behavior with the help of recommendation systems. This paper presents the surveying of the existing recommender systems, detailed view of the evolution, providing original taxonomy with reference to the energy

efficient recommendation systems. It includes the nature of the recommendation engine, computing metrics, evaluating platforms, and in-depth critical analysis. It also explains different strategies like conventional smart metering, edge computing, and privacy preservation recommendation. Recommending systems in any sectors is a very promising field in the energy consumption process. Table 1 summarizes on the approaches for power consumption with numerous ways of the recommendation system.

Table 1 Summary on the approaches for power consumption

S.no	References no	System/technology/stack used	Application
1	[4]	Hybrid machine learning approach for power forecasting	Renewable energy of fossil fuels
2	[6, 8]	Electricity plan recommender system	Retailing markets, electricity sectors
3	[9]	Moving recommendation and shift schedule recommendations	Commercial Buildings Power Usage
4	[1]	Energy prediction analysis and utilization models	Urban sectors on energy usage
5	[12]	The novel hybrid and ensemble prediction models	General purposes
6	[3]	Micro-moment recommendations	Automation services
7	[19]	Optimal task scheduling under dynamic electrical constraints	General purposes
8	[10]	Socio-technical gamification-based system	Web and mobile Interfaces
9	[18]	Peak demand management system	Electricity market sectors
10	[15]	Demand response scheduling System	power consumption on household devices
11	[16]	Consumption scheduling optimization mechanism	Home management with local energy sources
12	[5]	Personalized recommendation system	Recommendation Systems in all sectors
13	[17]	Location-based social network recommendation systems	Social networking
14	[7]	Large-scale social networks recommender Systems	General purposes
15	[2]	Energy efficiency recommender systems	Recommendation systems in all sectors

4 Conclusion

This paper presents a review of various power consumption systems based on different recommendation systems. There is a huge impact in the energy consumption of the residential users as they are not alerted on their regular usage. This paper investigates the individual residential power consumption and based on the analyzed data the power consumption alert indicator is triggered. Power Limit Indicator System (PLIS) helps in alerting the user with limit reached or limit exceeded metrics (type of service, no. of units consumed, no. of appliances used, etc.) which helps in reduction of the power consumption and in turn reduces the electricity tariff.

References

1. Khan PW, Byun YC, Lee SJ, Kang DH, kang JY, Park HS (2020) Machine learning-based approach to predict energy consumption of renewable and nonrenewable power sources. *Energies* 13:4870. <https://doi.org/10.3390/en13184870>
2. Eirinaki M, Gao J, Varlamis I, Tserpes K (2018) Recommender systems for large-scale social networks: a review of challenges and solutions, Elsevier. *Futur Gener Comput Syst* 78:413–418
3. Wei P, Xia S, Jiang X (2018) Energy saving recommendations and user location modeling in commercial buildings. UMAP18
4. Fraternali P, Cellina F, Herrera S, Krinidis S, Pasini C, Rizzoli AE, Rottondi C Tzovzas S (2018) A socio-technical system based on gamification towards energy savings. In: IEEE, 2018, PerSCC'18 - Workshop on pervasive sensing for sustainable smart cities and smart buildings
5. Sarwar BM, Karypis G, Konstan J, Riedl J (2002) Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering. In: Proceedings of the fifth international conference on computer and information technology, vol 1
6. Mohapatra SK, Barsocchi P, Mishra S, Tripathy HK, Bhoi AK (2021) A pragmatic investigation of energy consumption and utilization models in the urban sector using predictive intelligence approaches. *Energies* 14:3900. <https://doi.org/10.3390/en14133900>
7. Mosavi A, Bahmani A (2018) Energy consumption prediction using machine learning; a review. *Energies* 11(x). FOR PEER REVIEW
8. Sardianos C, Varlamis I, Chronis C, Dimitrakopoulos G, Himeur Y, Alsalemi A, Bensaali F, Amira A (2020) Data analytics, automations, and micro-moment based recommendations for energy efficiency. In: 2020 IEEE Sixth International conference on big data computing service and applications (BigDataService)
9. De Angelis F, Boaro M, Fuselli D, Squartini S, Piazza F, Wei Q (2013) Optimal home energy management under dynamic electrical and thermal constraints. *IEEE Trans Ind Inform* 9(3)
10. Corbellini A, Godoy D, Mateos C, Schiaffino S (2018) DPM: an ovel distributed large-scale social graph processing framework for link prediction algorithms. *Future Gener Comput Syst* 78:474–480
11. Bian D, Pipattanasomporn M, Rahman S (2014) A human expert-based approach to electrical peak demand management, IEEE. *IEEE Trans Power Deliv*
12. Paterakis NG, Erdinc O, Bakirtzis AG, Catalão JPS (2015) Optimal household appliances scheduling under day-ahead pricing and load-shaping demand response strategies. *IEEE Trans Ind Inform*. <https://doi.org/10.1109/TII.2015.2438534>
13. Zhu Z, Lambotharan S, Chin WH, Fan Z A game theoretic optimization framework for home demand management incorporating local energy resources. *IEEE Trans Ind Inform*. <https://doi.org/10.1109/TII.2015.2390035>

14. Sardianos C, Varlamis I, Chronis C, Dimitrakopoulos G, Alsalemi A, Himeur Y, Bensaali F, Amira A (2020) The emergence of explainability of intelligent systems: delivering explainable and personalized recommendations for energy efficiency, Wiley Periodicals LLC; Zhang Y, Meng K, Kong W, Dong ZY (2019) Collaborative filtering-based electricity plan recommender system. *IEEE Trans Ind Inform* 15(3)
15. Bao J, Zheng Y, Wilkie D, Mokbel M (2015) Recommendations in location-based social networks: a survey. Springer Science+Business Media, New York
16. Margaris D, Vassilakis C, Georgiadis P (2018) Query personalization using social network information and collaborative filtering techniques, *future Gener. Comput Syst* 78:440–450
17. He J, Li X, Liao L, Song D, Cheung WK (2016) Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In: Thirtieth AAAI conference on artificial intelligence
18. Himeur Y, Alsalemi A, Al-Kababji A, Bensaali F, Amira A, Sardianos C, Dimitrakopoulos G, Varlamis I (2021) A survey of recommender systems for energy efficiency in buildings: principles, challenges and prospects, *Elsevier. Information Fusion* 72:1–21
19. Berkovsky S, Taib R, Conway D (2017) How to recommend?: User trust factors in movie recommender systems. In: Proceedings of the 22nd International conference on intelligent user interfaces, ACM, pp 287–300

A Novel Hand Gesture Recognition for Aphonic People Using Convolutional Neural Network



S. Deepa , A. Umamageswari , and S. Menaka 

Abstract Hand gesture is a form of nonverbal correspondence wherein apparent activities are utilized to impart significant messages. Sign language is the most natural and effective way for communication between normal human beings and hearing impaired and aphonic people. Existing hand gesture recognition systems used k-means clustering algorithm and support vector machine algorithm that suffered from challenges such as poor edge detection and required trained persons to avoid diagnostic errors. The proposed architecture uses convolutional neural network for the recognition of hand gestures. Gesture sign is recorded and subjected to binarization where the image is separated into foreground and background. Contours are detected from the binarized image. Feature extraction is done using the Scale Invariant Feature Transform (SIFT) algorithm. The extracted features are given to the convolutional neural network classifier to recognize the hand gestures. The recognized hand gesture is given as output in the text format. The proposed architecture shows improved performance in terms of varying background and illumination conditions.

Keywords Hand gesture recognition · Binarization. SIFT algorithm · Convolution neural network

1 Introduction

Because of its wide range of applications, the hand gesture recognition system has got a lot of attention in recent years. Due to its applicability in a wide range of fields, sign language is one of the nonverbal communication methods that is gaining traction and developing a firm foundation detected earlier. The most common application is for people who are differently abled, such as the deaf and dumb. They can communicate with non-signing persons utilizing a hand gesture recognition system without the need for an interpreter. The most natural and effective means for hearing impaired and aphonic persons to communicate with each other is the sign language.

S. Deepa (✉) · A. Umamageswari · S. Menaka
Department of CSE, SRM Institute of Science and Technology, Ramapuram, Chennai, India
e-mail: deepas1@srmist.edu.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
R. J. Kannan et al. (eds.), *Computer Vision and Machine Intelligence Paradigms for SDGs*, Lecture Notes in Electrical Engineering 967,
https://doi.org/10.1007/978-981-19-7169-3_22

235

Gesture recognition is a primary role through sign language in communication. It is a quickly developing area inside computer vision and has drawn in critical examination because of its inescapable social effect. To address the challenges experienced by hearing impaired and aphonic people, it is imperative that a technology be developed that converts sign language into text so that they may communicate with normal people. To bridge the communication gap between hearing and speech disabled persons and the general public, a hand gesture detection system is necessary. The proposed architecture uses convolutional neural network for the recognition of hand gestures. Whang et al. [1] suggested using a radar sensor to identify hand gestures employing a time sequential inflated 3dimensions (TS- I3D) convolution neural network strategy for hand gesture detection based on frequency modulated continuous wave (FMCW) radar sensor. Maharani et al. [2] proposed a hand signal acknowledgment utilizing K-implies grouping and backing vector machine calculation which arranges and perceives the hand motions utilizing support vector machine calculation to further develop accuracy. Chanu et al. [3] proposed a Comparative Study for vision-based and information-based hand signal acknowledgment strategy utilizing Data glove based technique to perceive the hand motions. The proposed model was hard to use for useful purposes. Ahuja et al. [4] proposed a Static vision-based hand motion acknowledgment utilizing head part investigation. Arifulhoque et al. [5] proposed Computer vision-based motion acknowledgment for work area object control utilizing Baum–Welch calculation. Akmelia et al. [6] have fostered a continuous Malaysian gesture-based communication interpretation utilizing shading division and accomplished an acknowledgment pace of 90%.

Holden et al. [7] have proposed an Australian communication through signing acknowledgment framework, which utilizes mathematical shapes and position of the items to follow different objective articles and concentrate the hand motion elements to perceive the sign expressions. Starner et al. [8] have proposed a framework to perceive hand signals progressively and clarifies American Sign Language (ASL) utilizing the Hidden Markov Models (HMMs). The sign acknowledgment framework accomplished correctness somewhere in the range of 75% and close to 100%. Ren, Zhou et al. proposed Robust part-based hand signal acknowledgment with the Kinect sensor [9], which centers around fostering a section based, powerful hand motion acknowledgment framework with the Kinect sensor. Cooper et al., [10] proposed Sign language acknowledgment that specifies the critical parts of Sign Language Recognition (SLR). The constant persistent motion acknowledgment framework recommended by Liang and Ouhyoung [11] for gesture-based communication presents a wide jargon gesture-based communication. In this paper, the contribution to the Hand gesture recognition framework is given from the hand which joins binarization and shapes discovery for preprocessing and SIFT for highlight extraction. This calculation expects to perceive the hand gesture from a data set, for an info picture. The paper is depicted as follows. In Sect. 2, the proposed Hand gesture recognition algorithm is clarified. In Sect. 3, the procedure is carried out and results are acquired for different information bases, showing that it works acceptably. Conclusion and References are discussed in Sects. 4 and 5.

2 Proposed Methodology

The process that will take place in the paper is shown in the system architecture. The entire process is classified into two parts: training and testing shown in Fig. 1. Training is the actual process flow of the project. In the training part, the input video is taken and the binarization process will take place. The binarization is carried out using the thresholding algorithm. The thresholding is an image segmentation; it partitions the image extracted from the input video into foreground and background. The binarization step is followed by contour detection. Contours are the lines that connect all of the points along the image’s boundaries that have the same intensity. The find Contour () function is used to extract contours from the image. Next step is feature extraction. SIFT algorithm has been used for this. From the set of reference images, SIFT key points of the hand gesture image are first extracted and stored in the database and then the key points with quantitative information are furnished which can be used for hand-gesture recognition. After this, classification is done using the convolutional neural network (CNN) to recognize the hand gesture. To extricate and become familiar with a more significant level element of the hand motion picture, CNN applies a progression of channels to the crude pixel information.

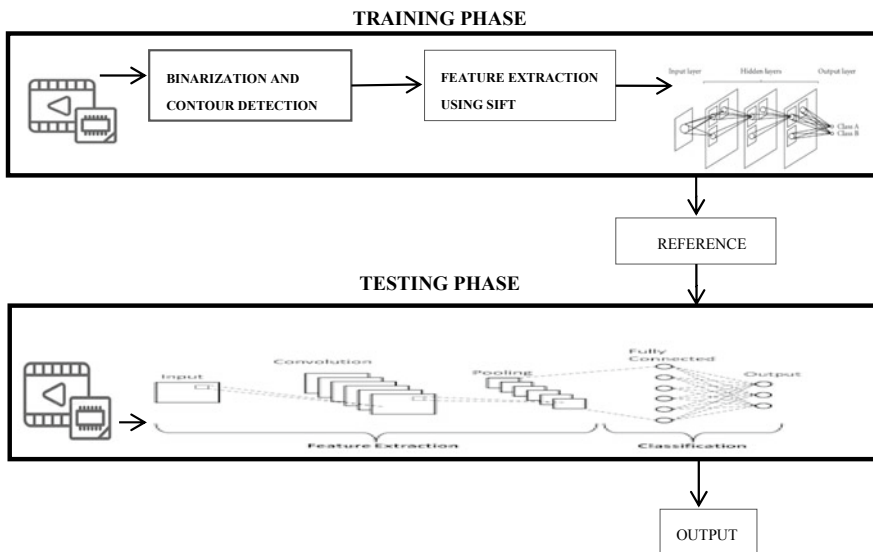


Fig. 1 Block diagram

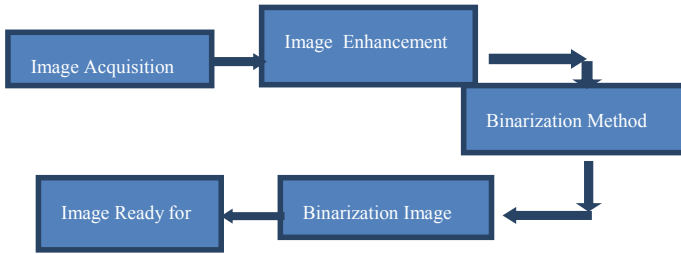


Fig. 2 Binarization model

Fig. 3 Binary image



2.1 Binarization

In general, all documents are saved in the computer memory as grey levels, with a maximum of 256 different gray values ranging from 0 to 255. Each gray value produces a unique color in the grayscale palette. If certain information from the document picture is necessary, the action must be processed multiple times. Binarization is a technique for turning any grayscale image into a black-and-white image. To begin the binarization process, the grayscale threshold value is determined and whether or not a pixel has a specific gray value as shown in Fig. 2 is also estimated. The output of the binary image is shown in Fig. 3.

2.2 Contour Detection

A contour is a curve that connects all points in the vicinity of a certain region of a picture that are of the same hue or intensity. The outlines of all items in the threshold image are identified to identify hand motions. Then there's the largest contour, which represents the hand's area and has the largest computed area. To find the hand contour, the Contour () method is utilized. The output of contour detection is shown in Fig. 4.

Fig. 4 Contours detected from binary image



2.3 Feature Extraction Using Sift Algorithm

The feature extraction phase follows the contour detection phase. Scale-invariant feature transform is the algorithm employed (SIFT). The SIFT algorithm extracts/generates image features that are invariant to picture translation, scaling, and rotation. The SIFT technique can be used to find distinct features in an image. The points of interest in the training/reference hand gesture images are detected and saved in the database, which are referred to as key points in the SIFT architecture.

The difference between successive Gaussian-blurred images is analyzed after the test hand gesture image is convolved using Gaussian filters at various scales. The Difference of Gaussians (DoG) maxima and minima at various scales are important sites.

2.4 Classification Using Convolutional Neural Network

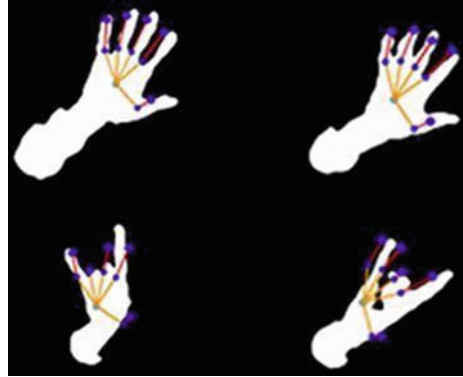
After the feature extraction phase, the classification procedure begins. Convolutional neural networks are utilized in this cycle (CNN). The structure of CNN is summarized in three layers.

Image Input Layer: The size of the input image is set in an image Input Layer, which in this case is 128-by-128-by-1. In this situation, the incoming data is a grayscale image, therefore, there is only one channel.

Convolutional layer: Convolution layers are a collection of filters that can be learned. The filtering size, number of filters, and padding are the input arguments for this layer. In this scenario, the size 10 filter is applied, resulting in a 10×10 filter.

Classification Layer: Classification layer is the final layer of the architecture. This layer classifies classes based on the probabilities acquired from the softmax layer, as well as calculating the cost function. The output is shown in Fig. 5.

Fig. 5 Hand gesture recognition using CNN bilateral filtering



3 Experimental Results

The input undergoes preprocessing steps to remove the noise. The recognized hand gestures are presented to the user in the form of text output.

We used Mendeley’s HGM-4 dataset to evaluate the performance of our suggested technique [12]. The HGM-4 dataset was created for hand gesture identification and contains a total of 4,160 color images (1280×700 pixels) of 26 different hand movements collected by four cameras at various angles. The original datasets utilized in this investigation were separated into three image sets: training, validation, and test, with proportions of 90, 7, and 3%, respectively. Table 1 shows the specifics of each of the three sets.

Accuracy: It is defined as the number of samples that are exactly predicted out of a total number of samples.

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

The True Negative is TN, the True Positive is TP, the False Positive is FP, and the False Negative is FN. Figures 6, 7 and 8 show the peace command, Loser command and Power command sign using Hand gesture recognition system. Figure 9 shows the CNN performance graph for training and validation. Table 2 gives the comparison of results with the existing system.

Table 1 The dataset is divided into three groups

	Total	Train (90%)	Validation (7%)	Test (3%)
Initial dataset	4160	3744	291	124

Fig. 6 Peace command recognized using Hand gesture recognition system



Fig. 7 Loser command recognized using Hand gesture recognition system

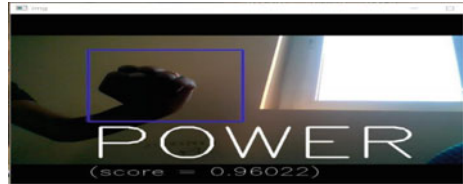


Fig. 8 Power command recognized using Hand gesture recognition system



Fig. 9 CNN performance graph

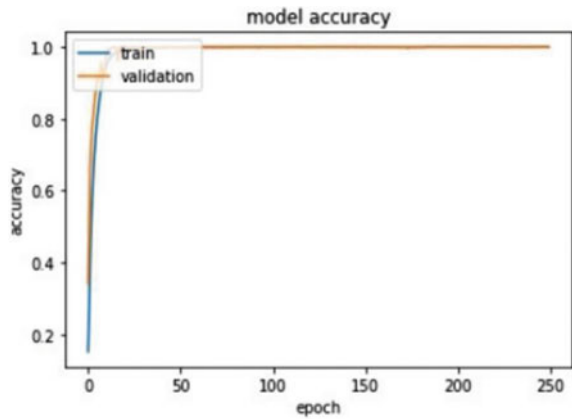


Table 2 Comparison of results with existing systems

Author, year	Methods	Language level	Accuracy
Shukor, 2014	Template matching	Alphabets, numbers Gestures	95.93.3 78.3
Sriram, 2013	Template matching	Alphabets	94
Matiwade, 2016	Logics level	Alphabets	93
Goval, 2013	Keypoint localization	Alphabets	95
Murakami, 1991	Neural networks	Alphabets, words	98
Pail, 2014	Template matching	Alphabets	96
Lu, 2016	Extreme learning machine	Numbers	91.2
Xie, 2012	Segmentation algorithm	Basic gestures Complex gestures	97.9 97.2
Wu, 2015	Libsvm	Words	95.9
Shasha hang, 2019	Ts-i3d	Gestures	97
Anggi, 2018	K-means clustering and SVM	Gestures	98
Proposed system	Proposed method	Gestures	99

4 Conclusion and Future Enhancement

This paper addressed the problem faced by the hearing impaired and aphonic people in our society. We have proposed a system that helps hearing impaired and aphonic people to communicate with the normal people. The system classifies the hand gestures using convolution neural network. Because the system may be used in a mobile situation, it's ideal for real-time applications. Research on vision-based hand gesture identification is currently ongoing, and our future research will focus on increasing the accuracy.

References

1. Whang et al (2019) TS-I3D based hand gesture recognition method with radar sensor
2. Maharani et al (2018) Hand gesture recognition using K-means clustering and support vector machine
3. Chanu et al (2017) Comparitive study for vision based and data based hand gesture recognition technique
4. Ahuja et al (2015) Staticvision based hand gesture recognition using principal component analysis
5. Arifulhoque SM et al (2018) Computer vision based gesture recognition for desktop object manipulation
6. Akmeliawati R, Ooi MPL, Kuang YC (2007) Real-Time Malaysian sign language translation using colour segmentation and neural network. In: IEEE on instrumentation and measurement technology conference proceeding. Warsaw, Poland, pp 1–6
7. Holden E-J, Lee G, Owens R (2005) Australian sign language recognition. *Mach Vis Appl* 16(5):312–320
8. Starner T, Pentland A (1995) Real-time American Sign Language recognition from video using hidden markov models. In: International symposium on computer vision, pp 265–270
9. Zhou et al. (2013) Robust part-based hand gesture recognition using kinect sensor. *IEEE Trans Multimedia* 15(5):1110–1120
10. Cooper H, Holt B, Bowden R (2011) Sign language recognition. In: *Visual analysis of humans*. Springer London, pp 539–562
11. Liang R-H, Ouhyoung M (1998). A real time continuous gesture recognition system for sign language. Automatic face and gesture recognition. In: Third IEEE international conference
12. <https://data.mendeley.com/datasets/jzy8zngkbg/4>

Comprehensive Analysis of Defect Detection Through Image Processing and Machine Learning for Photovoltaic Panels



S. Prabhakaran, R. Annie Uthra, and J. Preetharoselyn

Abstract Fault identification in Photovoltaic (PV) panels is of prime importance during the regular operation and maintenance of PV power plants. An extensive fault identification process that employs Image Processing, Machine Learning, and Electrical-based techniques has been analyzed comprehensively. Photovoltaic panels are the perfect choice of renewable energy from natural light sources. The energy yield of PV panel is degraded gradually because of dust, discoloration, crack and faults. These faults attribute a production loss of up to 20%, as one faulty panel can bring down the efficiency of the entire plant. It is implied that faulty panels need to be identified as soon as possible, and it is highly critical to resolving issues to ensure effective energy production. The various IP techniques and ML models that elucidate ways to identify PV panel defects have been addressed. Inferences made from this study to help identify three methods for defect detection that stand apart in terms of efficiency. Parametric observations on all three methods are made in terms of F1 Score, Precision, Accuracy and Recall. Numerical values indicate clearly that the Machine Learning method based on AlexNet predicts the Photovoltaic panel's defect with 0.86 F1 value, 0.89 precision and an accuracy of 85.56%.

Keywords Photovoltaic panel · Image processing · Machine learning · Deep learning

1 Introduction

Renewable energy resources gave the biggest addition to power generation, followed by gaseous petrol when the coal age fell. The portion of renewable energy in power generation expanded from 9.3 to 10.4%, outperforming atomic energy. There is

S. Prabhakaran (✉) · R. Annie Uthra · J. Preetharoselyn
SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India
e-mail: sbaprabhu@gmail.com

R. Annie Uthra
e-mail: annieu@srmist.edu.in

Primary Energy Consumption by energy source, 2020

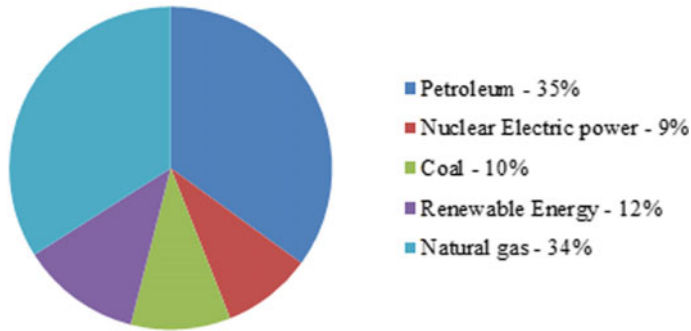


Fig. 1 Primary energy consumption by energy source

boundless help for utilizing sustainable power, especially sunlight-based energy, geothermal energy, wind energy and other types of biomass energy, which give energy without any carbon dioxide emissions. According to REN21's 2017 study, renewable energy accounted for 19.3% of global energy consumption. This energy is divided as follows: 8.9% from traditional biomass, 4.2% from heat energy, 3.9% from hydroelectricity and 2.2% from wind, solar-powered, geothermal, and other various forms of biomass. In 2015, global interest in environmental growth amounted to more than US\$286 billion. Wind and solar power become problematic at substantial levels while implemented [1]. Consumption of primary energy is usually classified as depicted in Fig. 1.

1.1 Photovoltaic Panels

The energy crisis has become a big issue since the beginning of the twenty-first century. Worries are raised about settling natural contamination issues, cruel atmosphere changes and exhaustion of fossil fuel sources. Subsequently, the innovation of PV force has been quickly developed [2]. In reality, popular fuel sources, for example, atomic, oil and coal, affect the biological system, human well-being and environmental change [3]. This crisis led to the development of renewable energy sources, mainly solar energy; this is our only focus. The characteristics of solar power are pure, inexhaustible, viable and other organic benefits which create new energy. PV plants are becoming increasingly widespread across the world every year. Generally, we used Monocrystalline and Polycrystalline types of PV cells [4].

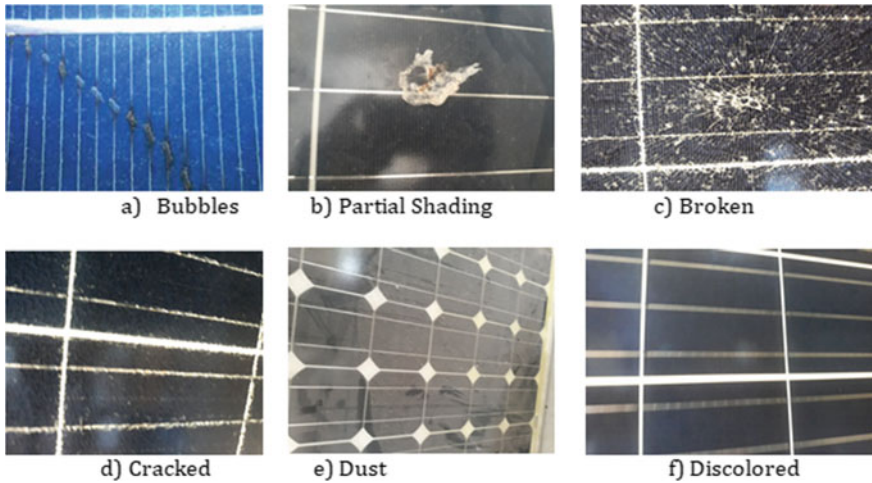


Fig. 2 Real-Time images of defective photovoltaic modules

1.2 Defects in Photovoltaic Panels

The faults in PV panels consist of different sizes and shapes. To segment the micro-cracks in high-definition images, we need a profoundly adequate and efficient methodology. Small industries often utilize human-inspected fault identification arrangements. Human-inspected fault identification strategies are inadequately productive and costly. Programmed fault identification techniques were developed by Image Processing and Machine Learning, which have rapid, better accuracy and minimal effort [4]. Some of the most widely recognized PV panel deformities such as blob, yellowing, browning, busted module, fractured module, rust, charred components, separation, breakable, lax, crouched, incorrect positioned, exposed electrical parts, oxidized, delaminated, corrosion and anti-reflection snail trails are required to keep a watch on during PV installation. We have collected some real-time PV panel images having different kinds of faults which are depicted in Fig. 2. and described below in Table 1.

2 Fault Identification System

Fault identification and classification system provides high efficiency and good production. This system provides the root cause and location of the particular defect. Fault identification system is based on mathematical models and process modeling. These models are associated with statistical decision-making techniques, expert systems, artificial neural networks, fuzzy logic and computational procedures. So this fault identification system becomes more effective and completely independent

Table 1 Defects versus nature of severity on solar panels

Discoloration	The Silver grid across the PV module is described as Optical discoloration. The discoloration is caused by internal and external circumstances such as poor temperature and quality of PV cell [5]
Cracking	Micro-cracks occur during manufacturing, manual soldering or by applying mechanical and thermal stress on it. In continuous progress, cracks are expanded largely which can reduce the module yield [5]
Anti-Reflection coating damage	Due to dirt and rust retention, anti-reflection coating damage and production loss may occur [5]
Soiling	Soiling can reduce the panel’s work over a long period and it is caused due to dirt and bird dung [5]
Busbar oxidation and corrosion	One of the primary degradation methods in PV method is encapsulant. It takes place between PV cell’s exterior and encapsulant. Modest components and mistake-handling activities bring about improvements in shading [5]
Hotspots	It may also occur due to the PV panel’s inherent properties, such as electrical fluctuations, impurities and wafer resistance [5]
Physical impacts	Serious physical impacts are made on surface of the PV panels. They occur due to climate, misuse of components upon migration, significant thermal extension and glass breakage. Previously mentioned deformities influence the yield of the PV module significantly. A few deformities can be prevented by improving assembling measures, utilizing better materials and more cautious dealing with components [5]

in real-time operation. In Fig. 3, a block diagram is introduced where the consecutive design of the fault identification system is described. The various fault identification systems are utilized using different techniques which are depicted in Fig. 4.

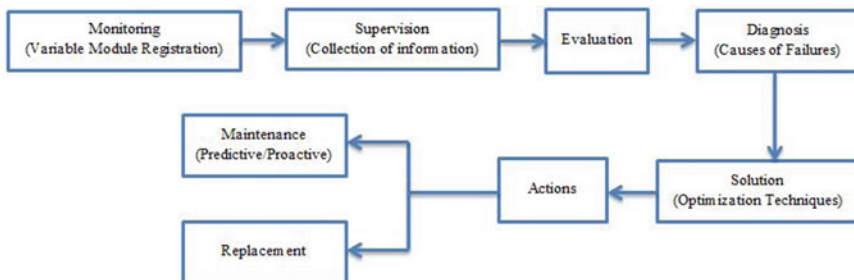


Fig. 3 Common workflow diagram of fault identification system

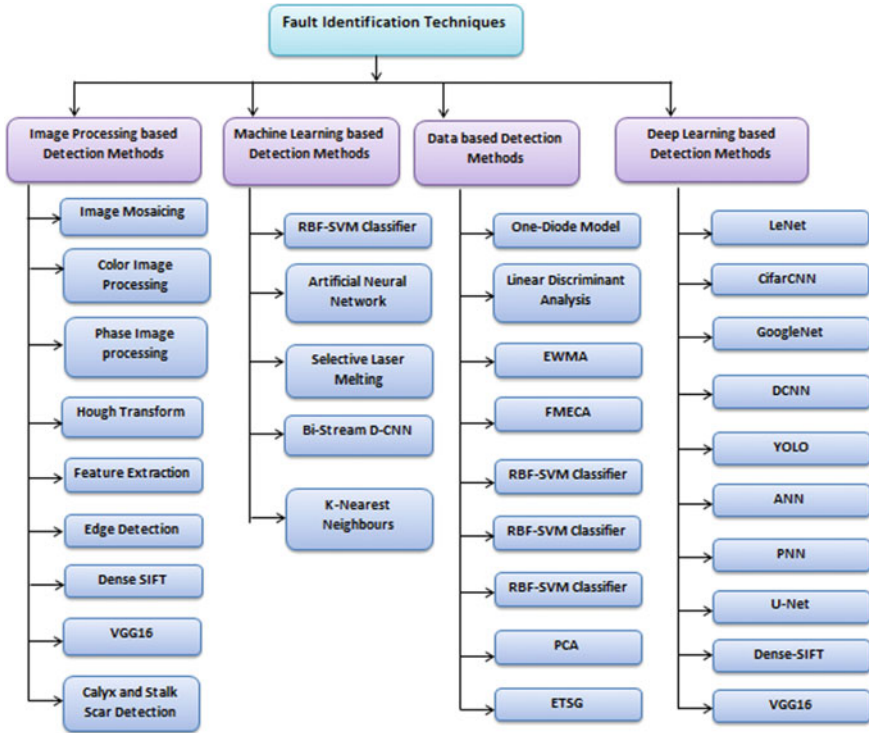


Fig. 4 Overview of fault identification methods

2.1 Image Processing-Based Defect Identification

Image processing defect identification system performs with various procedures which allow pixels to be classified as defects and non-defects [6]. As a result, only the image-relevant data is extracted and the rest is ignored. The Image Processing techniques build up information preparing work process that consistently and consequently perform basic pre-processing and post-processing image processing steps shown in Fig. 5.

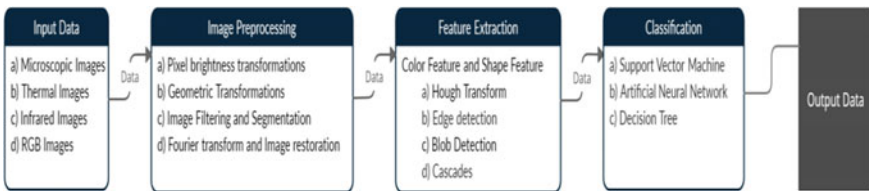


Fig. 5 Common Workflow diagram in image processing

To identify the micro-cracked faults in PV panels, image processing techniques and Electroluminescence innovation techniques were used. To achieve the image segmentation threshold, the OTSU algorithm is used and to discover tops in the parameter space dot-line duality [7]. It depends on the feature vector that describes parts of images acquired by a standard camera. It relies on the measurement of co-occurrence matrices for images. OR-Login activity is performed between the two co-occurrence matrices, and the segmentation technique is applied once a PV panel pattern has been characterized. Initially, the image is investigated and then separated into different partitions. Then the feature vector is determined for every partition. At last, vector probability is calculated for the panel class [6].

From the below numerical equation, Variance can be explained as the distribution of the data

$$\sigma^2 = \frac{\sum_{i=0}^n (X_i - \mu)^2}{N}$$

where

X_i —Pixel Value

μ —Mean

N —Number of pixels in one image

An image processing methodology is used for improving the activity of a PV plant regarding dust evaluation. The procurement of images is performed by drones outfitted with high-resolution cameras. The calibration technique is used to create a library of dustiness images in instances of various lighting levels such as Low lighting, Medium lighting and High lighting. The main innovation of this algorithm is used to detect dust and to measure the dust amount on a surface. The outcome of the first method depended on the average ratio between the grayscale level pixels, and the outcome of the second method depended on regression analysis [8]. Infrared thermography-based defect identification framework was created to identify some expected faults in PV boards. To identify the defects in PV panel images, feature extraction and classification algorithm were used. Background data are gotten rid of by canny edge detection and Hough transformation. The overall Training efficiency = 93.7%, Validation efficiency = 96.3% and the testing efficiency = 93.3% [9]. Harris Corner identifier technique is used to consolidate multiple Infrared images to Image mosaicking. In a PV factory, a small Unmanned Aerial System—PLP-610 is used to identify the faults in PV panels. The blending algorithm is utilized after the mistakes of photometric and geometric misalignment between images given by the stitching phase. Hence, the blending algorithm is used to limit the breaks in mosaic images and to check entire defect PV panels and safe panels in PV boards [10]:

$$f(a.b) = \sum [I(a_i - b_i) - I(a_i + \Delta a, b_i + \Delta b)]^2$$

where

$f(a,b)$ —window function

$a_i - b_i$ —shifted intensity

An automatic wafer pattern recognition framework is used for enhancing the efficiency, fault classification functions and removal of noise, etc. Compared to manual auditing, multilayered wafer patterns are 30 times faster than human auditors. The classification data is used to indicate the audit zone, as well as to separate defective areas into multiple levels. An input image signal is analyzed by a rapid pipeline-structured image processor at a rate of 7 MHz in a one-pass manner. Accuracy of the detection framework is adequate because the prediction error is less than 0.5 occupancies per chip [11]. This algorithm with microscopic imaging is to identify mathematical flaws like an incomplete filling, harmful designs and sticking in printed patterns. Nano-Imprint Lithography is a powerful technique for the production of huge Nano-structures at low prices. The fundamental issue of microscopic defect detection was resolved by morphological processing and linear filtering. Across 102 defects, the biggest defect size in a single image is $7 \mu\text{m}^2$ and the average size is $2 \mu\text{m}^2$. Overall value of the calculated faulty area is 2% [12].

In [13], a Phase Unwrapping Algorithm is developed to identify glass defects such as stone, knot and ream. This algorithm depends on Phase image processing. To eradicate jump error, this algorithm is used which depends on the jump-corrected method. The division of the faulty area is carried out by threshold segmentation and grayscale mathematical morphology. Boundary coordinate is used to measure the fault's proportion and position. This phase unwrapping algorithm is utilized to accomplish the actual algorithm rapidly and accurately and will ensure the accuracy of every fault and faulty area. Instead of grayscale image processing, Color Image Processing [14] is used to create image sensors. The proposed rust fault identification technique can be additionally evolved to understand a robotized mechanical investigation of bridge coating surface conditions Table 2.

2.2 Machine Learning-Based Defect Identification

Most numerical classification methods depend on supervised learning models. These classification methods are trained with benchmark datasets to identify the faults and defects accurately. There are various detection and classification techniques developed for fault identification problems. The traditional technique called K-nearest-neighbors algorithm (kNN) is used to identify the defects in PV Panels. Even instance-based algorithm has some issues when it is utilized on enormous datasets. Data collection, data pre-processing, building datasets, model training, testing evaluation and deployment to production are all common phases in Machine Learning. Some sections of the machine learning operations can be automated. It covers data pre-processing, data cleaning and feature development, as well as how they affect Machine learning Model performance [18]. Machine learning workflows

Table 2 Advantages and disadvantages of defect detection and classification with specific algorithms

Input Data	Algorithm/techniques	Advantages	Disadvantages
Infrared image [9, 10]	Unmanned aerial system Image mosaicking	Resultant image can also be used for texture mapping of a 3D environment	Multiple images cannot be achieved repeatedly May fail if sequence id is missed
Thermal image [8]	Feature extraction Edge detection Though transform	External tracking Least detection time Strong efficiency of classifying	Sensitivity and resolution reduce with distance and angle of view
CAD data [8]	Pipeline architecture Image processing techniques	Specifies the Field of Inspection Designation of optimum inspection parameters Differentiate faulty parts into various layers	Complex Compilation Inability to continuously run the pipeline at the full speed
Ordinary image [15]	Machine vision RBF-SVM classifier Calyx and stalk scar detection	Good generalization Easy to design Works well even if the dataset is not linearly separable	Low precision Labor-intensive Subjectivity
Ordinary image [16]	Bi-stream deep convolutional neural network Selective laser melting	Easily identifies trends and patterns No human intervention is required	Hyper parameter tuning is not trivial Have to enhance the consistency and repeatability of the component
Ordinary image [13]	Phase image processing	Clear edges with good segmentation results	Not suitable for noisy images Not suitable for edge-less images
Ordinary RGB images [14]	Color image processing	A Accuracy, objectivity Speed, consistency High adaptability to changes in pixel values without losing precision	Highly redundant and correlated
Ordinary Image [15, 17]	Feature extraction DCNN, faster R-CNN Dense-SIFT, VGG-16	DCNN Concise, compact, easy to use Faster classification Automatic positioning	Complex process Depends on region selection algorithm

(continued)

Table 2 (continued)

Input Data	Algorithm/techniques	Advantages	Disadvantages
Microscopic Images [12]	Image processing Nano-imprint lithography	Simple equipment setup	Erase the pattern image difficulty in printing large patterns and complex patterns

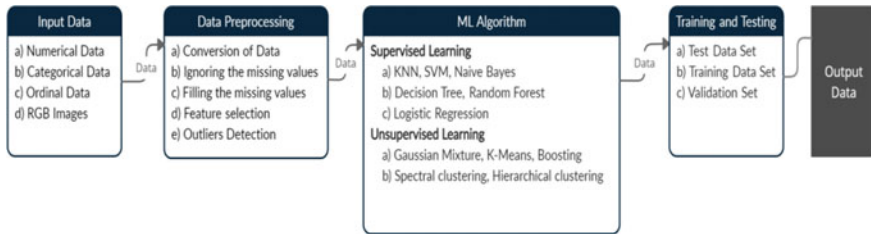


Fig. 6 Common workflow in machine learning

define typical phases including data collection, data pre-processing, building datasets, model training and refinement, evaluation and deployment to production which are depicted in Fig. 6.

Cascaded Forest identification system will find four types of PV field failures such as short circuit, open circuit, abnormal degradation and partial shading. To find out the failures in PV panels, a Cascade Forest model can be prepared, and afterward, real-time information can be sampled. Comparing the Random Forest and Back Propagation Neural Network (BPNN) with test information, we can check the reliability. The test results show that Cascaded Forest is superior to the Random Forest model and the BPNN model [2]. BPNN-based defect identification for PV panels, can show the connection between the PV array faults and the reason for the shortcoming. The proper defect patterns such as battery breaking, short circuit and shadow are used to train the neural network. Using various parameters such as temperature, voltage and current, the PV panel’s faults can be identified. To train the network, this experiment collected 200 groups of fault sample data under various conditions. Using the Tansig Transfer Function, a neural network was trained, the training error was 0.001 and the training phase duration was 0.05 s per dataset [19].

Generally, Machine Learning methods are dependent on RGB images and identify some external faults such as mud, snow and sand in PV panels. The improvement of this technique depends on data gathered from the PV module control framework. A Machine Learning pipeline was created for observing PV panel status by utilizing information obtained from various sources. For the prediction of future defects, the Machine Learning methods use binary classification. Totally, this Machine Learning pipeline has 342 datasets for training and testing, and the result score got is 0.9 in LBFSGS and LIBLINEAR classification algorithm [20]. The CALYX and STALK SCAR identification process is performed by histogram thresholding and mean

Table 3 Accuracy rate of different Methodologies

References	Defect detection techniques	Accuracy rate (%)
[2]	Cascaded forest	85
[2]	Random forest	84
[21, 22]	YOLOv3	91
[21, 22]	YOLOv2	89
[19]	Tansig transfer function	90
[3]	PNN classifier	98.19
[23]	ETSG	90
[17]	Dense-SIFT	99.26
[17]	VGG-16	97.14
[4]	U-net	97.38
[15]	RBF-SVM	97.09
[9]	Infrared thermography	94.0

grading-recognition (g-r) estimation. RBF-SVM classifier utilizes the Lab color space pixel values to identify the faulty locations [15]. Deep Convolutional Neural Network (DCNN) analyzing SLM, and self-regulating image feature learning and feature integration techniques are accomplished in SLM faulty condition-related patterns [16] Table 3.

2.3 Deep Learning-Based Defect Identification

Most of the shallow learning models extract a few feature values from signals, causing a dimensionality reduction from the original signal. Deep learning methods used complex layers with new classification methods to identify the faults in PV panels. Most of the machine learning models extricate a couple of elements from actual data. So the dimensionality reduction may happen from the original data. A Deep Learning computer model figures out how to perform classification operations directly from pictures, text or sound. By using CNN, the data can be classified into normal and faulty classes. Such a method tries to look at all issues, and it provides better results [24]. A traditional deep learning workflow is depicted in Fig. 7.

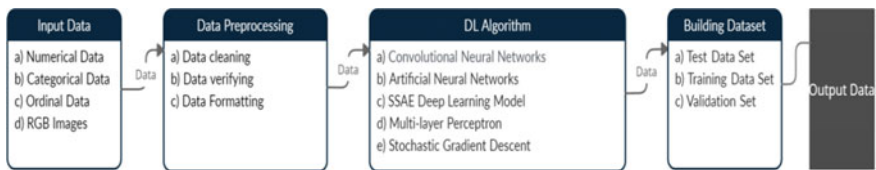


Fig. 7 Common workflow in deep learning

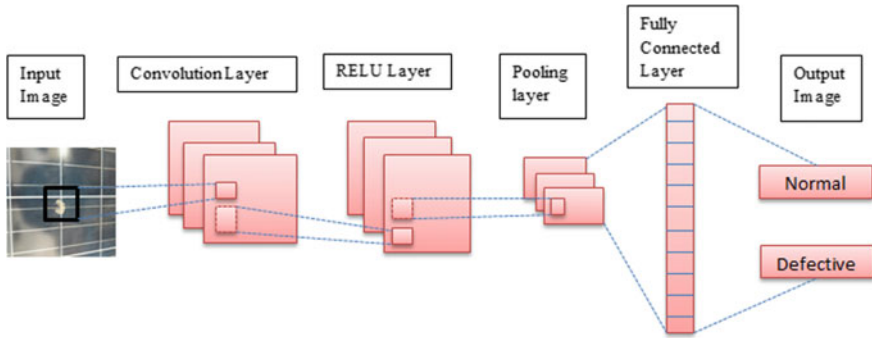


Fig. 8 Common workflow in convolutional neural network

Deep Learning-based CNN classification methods such as LeNet, CifarCNN, YOLO and GoogleNet are used to distinguish PV cells as either damaged cells or ordinary cells. Here, we used binary classification for fault identification and the training algorithm is Stochastic Gradient Descent. The learning algorithm was also optimized using root mean square propagation [1]. An Unmanned Aerial Vehicle (UAV) system configured with the CNN YOLOv3 technique was used to find the PV Panel defects like white stains, dust, glass breakage, grid-line corrosion, snail trails and yellowing. To identify the faults in each cell, an Image processing-based segmentation method was used. The proposed technique is compared with previous fault identification methodologies with a feasible dataset. Principle advantage of utilizing DCNN-based YOLOv3 is it will give all the data needed to discover the objects in the images. The YOLOv3 defect identification method accomplishes a mean average precision of 98.4% [21, 22]. The Common architecture of CNN is depicted in Fig. 8.

A UAV system configured with the DCNN technique was used to find and classify the defects in Electroluminescence PV panel images. Some of the defects are dust shading, encapsulant delamination, grid-line erosion, snail trails and yellowing. The U-net model is prepared and tried on solar Electroluminescence images dataset, and the outcomes are recorded utilizing mean Intersection over Union. It altogether improves the effectiveness and precision of resource review and well-being appraisal for PV farms with the ordinary methods. This algorithmic solution is broadly assessed from various angles, and the mathematical outcome unmistakably exhibits its viability for the productive fault identification of PV panels [4, 25]. The contracting path of U-net is as follows.

W. Chine et al. proposed an Artificial Neural Network (ANN)-based defect identification technique for five different kinds of defects such as short circuit fault, inversed bypass diode fault, shunted bypass diode fault, open circuit fault and connection resistance between PV modules. Also, this detection algorithm is used to predict the Alternating Current (AC) power creation. By using a simulation model, PV panels are calculated in given working condition parameters like temperature, current, voltage and a number of peaks in the current–voltage. To produce day-by-day predictive

alarms and to distinguish faults, pattern information of PV panels is compared with model performance, and the vector of residuals is analyzed. The auto-correlation work is then used to discover the characteristic periodicity of peaks. The residuals are collected and handled to identify out-of-limit tests and framework degradation trends; these trends are separated by processing the Triangular Moving Average. The method has been approved utilizing an exploratory information base of climatic and electrical parameters [23, 26]. The consecutive current and voltage of the PV panels are changed into a Two-Dimensional Electrical Time Series Graph (ETSG). This design consists of two primary parts such as feature extraction and classification. The above design consequently separates appropriate features from ETSG, which gets rid of the need to utilize artificially settled features of information. This defect identification technique just takes the input (voltage and current) of the PV panel and the reference panels are utilized for normalization [27].

In [3], a Probabilistic Neural Network (PNN) defect identification and examination technique is used to identify the short circuit faults. The recommended methodology comprises six principal phases such as parameters extraction, elaboration of database, network construction, training, testing, simulation and validation. The PNN technique was compared with feed-forward back-propagation and ANN classifier technique, for silent and faulty information. In [17], a fastener fault detection technique utilizes Dense-SIFT and VGG-16. The outcome is conceivable to complete the imperfection recognition of fasteners with RCNN. This methodology is applied in railway track fault identification such as fastener missing and broken cases. To identify the faults, here we used feature extraction and classification. The key to the Dense-SIFT algorithm is to determine the descriptors of each bin. They are evaluated here as follows

$$(a, b) = \sqrt{(L(a + 1, b) - L(a - 1, b))^2 + (L(a, b + 1) - L(a, b - 1))^2}$$

$$\theta(a, b) = \arctan \frac{L(a, b + 1) - L(a, b - 1)}{L(a + 1, b) - L(a - 1, b)}$$

where

$L(a,b)$ —input image

$\theta(a,b)$ —Gradient magnitude

3 Experimental Results and Analysis

The comprehensive survey is supported by experimental and numerical results obtained with datasets that contained thermal imageries. Each classification model is chosen and the best model is identified for obtaining experimental results.

3.1 Canny Edge Detector

Canny Edge detection with Optical Flow, followed by DBSCAN, has been identified as the best method for detecting defects in solar panels. The biggest benefit is that this method is suitable for real-time applications due to its low computational cost. It also has good detection rates and close to zero false alarm rates, and finally, is resilient to motion blur. Also, this method can detect the most common problems—cracks and hotspots immediately when it comes into the field of view of the camera, while hot panels will be clustered using DBSCAN. Also, this method uses both pixel-level scanning and global scanning to give the output, which makes it even better. The Canny edge detector depends on processing the squared gradient magnitude. The arithmetic behind the entire improvement measure is somewhat complicated. By using a Gaussian low-pass filter and gradient magnitude, the images were smoothed and the edges were detected. The Gaussian low-pass filtering is used to compute the gradients significantly, and it can reduce the noise sensitivity of the edge detector. The Canny edge detection algorithm can be summed up below.

Algorithm 1: Canny Edge Detector

Step 1: Gaussian filter to be applied for smoothen the image.

Step 2: Intensity gradients need to be finding for an image.

$$|G| = \sqrt{I_x^2 + I_y^2}, \theta(a, b) = \arctan\left(\frac{I_x}{I_y}\right)$$

Step 3: On-maximum suppression to be applied to edge detection

$$p = \alpha a + (1 - \alpha)a$$

Step 4: Double threshold image binarization to be applied to determine solid edges.

$$highThreshold = \max(\max(im)) * highThresholdRatio;$$

$$lowThreshold = highThreshold * lowThresholdRatio;$$

Step 5: Track edges by hysteresis.

The input comes from a machine scanning panels in sequence. The output is a detection of various types of defects as stated before—cracks, hotspots and hot panels. The experimental results are as follows: When tested on real-world test data from close to 12,000 solar panels, accuracy was 98% of panels correctly recognized. And from these, 92% of all types of defects were detected by the system.

3.2 Support Vector Machine

The best method out of 5 methods outlined for Machine Learning-based methods is Anisotropic filtering with an SVM classifier. Electroluminescence imaging is used for capturing PV images. This particular combination of methods performs the task of micro-crack detection really well because the diffusion methods produce excellent enhancement. The SVM can perform efficient classification. This means that the method will work even with various forms of noise.

240 8-bit sample images of resolution 1178×1178 are taken as input for defect detection. The output classifies whether micro-cracks are present or not. The Classification rate for SVM is 88% with average sensitivity = 97.7% and specificity = 80.2%.

Figure 9 provides the input and output as a result of experiments conducted on PV panels using Anisotropic filtering along with the SVM classifier. The panels with hotspots are identified for maintenance thus preventing an avalanche effect from occurring in the future.

From the above graphs, it is evident that AlexNet outperforms the other three methods in terms of 3 parameters, namely F1, Recall and Accuracy. This clearly indicates that the best possible method for detection of defects in solar models is through Machine Learning.

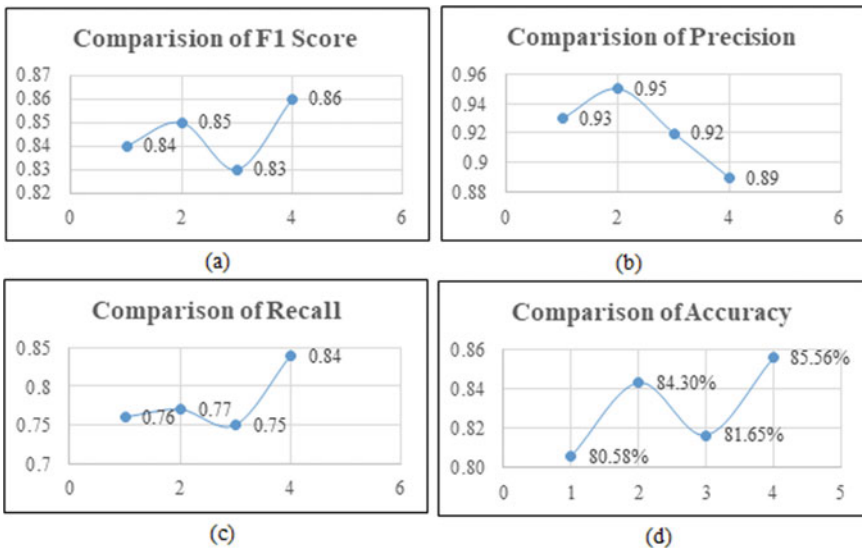


Fig. 9 Comparison of parametric observations for all 4 methods—**a** F1 Score, **b** Precision, **c** Recall, and **d** Accuracy

3.3 AlexNet

Of all the methods available, the best method for solar panel defect detection is AlexNet. It is a 25-layer Feed-Forward CNN. The image type is Electroluminescence imaging. Broadly, there are two categories of Deep Learning algorithms that can be applied here—Classification and Segmentation algorithms. Among these, classification algorithms using CNN are the better option because they provide automatic feature extraction and classification as well. Also, transfer learning applied on AlexNet provides the best results according to the paper. The dropout with another good candidate VGG-16 is that it suffers in handling low-resolution images.

The Input is a $227 \times 227 \times 3$ image and the output is the category that the picture belongs to—either defective or normal. Therefore, this method cannot detect between different categories, but this can simply be resolved by giving different categories of defects in the dataset without any change to the underlying architecture. A Dataset of 599 images (326 defective, 273 normal) from Google, Bing, etc. is taken into consideration. The Images are resized to $227 \times 227 \times 3$. The Validation accuracy stands at 93.3%.

The performances of deep learning algorithms are evaluated by Precision, Recall, F-Score and Accuracy. Those values are calculated by True Positives (TP), False Positives (FP) and False Negatives (FN), with the following formulas:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

where

P—Precision, R—Recall, and F—F1 Score.

Table 4 provides a clear picture of which of the four models provides the optimal result in the detection of PV Panel defects. Numerical results prove that the Deep Learning model using AlexNet outperforms other approaches. Deep learning model provides automatic ways to perform feature extraction while in other methods; such extraction is handcrafted or highly customized. Therefore, they do not generalize well. Deep learning can generalize to multiple scenarios, and also adding different types of cracks will require little to no architectural changes.

Table 4 Parametric evaluation of edge detection, SVM classifier, PCA and AlexNet

Method	F1 score	Precision	Recall	Accuracy (%)
Edge detection + Optical flow	0.84	0.93	0.76	80.58
SVM classifier	0.85	0.95	0.77	84.30
PCA+ ICA	0.83	0.92	0.75	81.65
AlexNet	0.86	0.89	0.84	85.56

4 Conclusion and Discussions

In this survey, there are various kinds of methods which have been proposed to identify the faults in PV panels. These methods can identify the faults quickly and effectively. We used parameters such as current, voltage, temperature and faulty image types such as IR images, Thermal images, RGB images and Microscopic images. These techniques are classified into three categories: (i) Image Processing-based defect identification, (ii) Machine Learning-based defect identification and (iii) Deep Learning-based defect identification. The comparison table shows various diagnostic algorithms and input factors. The developments of fault identification systems are most important in PV modules. Among the above techniques, the DCNN-based method gives better accurate results. Therefore, image processing and traditional Machine Learning methods will always fail to generalize to new types of defects and will require retraining and more handcrafting. Deep learning can learn the features automatically with sufficient data. This qualifies as the best candidate for defect detection in Solar panels.

References

1. Banda P, Barnard L (2018) A deep learning approach to photovoltaic cell defect classification. In: Proceedings of the annual conference of the South African Institute of computer scientists and information technologists on (SAICSIT). Port Elizabeth, South Africa, pp 215–221
2. Hu L, Ye J, Chang S, Li H (2017) A novel fault diagnostic technique for photovoltaic systems based on cascaded forest. In: Proceedings of the workshop on smart internet of things—SmartIoT 17. ACM Press, pp 1–5
3. Garoudja E, Chouderb A, Kara K, Silvestre S (2017) An enhanced machine learning based approach for failures detection and diagnosis of PV systems. *Energy Convers Manag* 151:496–513
4. Muhammad RUR, Chen H, Xi W (2019) U-Net based defects inspection in photovoltaic electroluminescence images. In: IEEE international conference on big knowledge (ICBK). IEEE, pp 215–220
5. Djordjevic S, Parlevliet D, Jennings P (2014) Detectable faults on recently installed solar modules in western australia. *Renew Energy* 67:215–221
6. Salamanca S, Merchán P, García I (2017) On the detection of solar panels by image processing techniques. In: Proceedings of the 2017 25th Mediterranean conference on control and automation (MED), Valletta, Malta, pp 478–483

7. Xu P, Zhou W, Fei M (2015) Detection methods for micro-cracked defects of photovoltaic modules based on machine vision. In: IEEE international conference on Cloud Computing and intelligence systems. IEEE, pp 609–613
8. Qasem H, Mnatsakanyan A, Banda P (2017) Assessing dust on PV modules using image processing techniques. In: IEEE 44th Photovoltaic specialist conference (PVSC). Washington, DC, USA, pp 1–5
9. Kurukuru VSB, Haque A, Khan MA (2019) Fault classification for photovoltaic modules using thermography and machine learning techniques. In: Proceedings of the International conference on computer and information sciences. Aljouf, Saudi Arabia, pp 1–6
10. Aghaei M, Leva S, Grimaccia F (2016) PV power plant inspection by image mosaicing techniques for IR real-time images. In: Proceedings of IEEE 43rd photovoltaic specialists conference. IEEE, pp 3100–3105
11. Yoda H, Ohuchi Y, Taniguchi Y, Ejiri M (1988) An automatic wafer inspection system using pipelined image processing techniques. IEEE Trans Pattern Anal Mach Intell 10(1):4–16
12. Pietroy D, Gereige I, Gourgon C (2013) Automatic detection of NIL defects using microscopy and image processing. Microelectron Eng 112:163–167
13. Jin Y, Wang Z, Chen Y, Kong X, Wang L, Qiao W (2014) Study on inspection method of glass defect based on phase image processing. Optik 125(19):5846–5849
14. Lee S, Chang LM, Skibniewski M (2006) Automated recognition of surface defects using digital color image processing. Autom Constr 15:540–549
15. Ileri D, Belal E, Okinda C, Makange N, Ji C (2019) A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing. Artif Intell Agric 2:28–37
16. Caggiano A, Zhang J, Alfieri V, Caiazzo F, Gao R, Teti R (2019) Machine learning-based image processing for on-line defect recognition in additive manufacturing. CIRP Ann Manuf Technol 68:451–454
17. Wei X, Yang Z, Liu Y, Wei D, Jia L, Li Y (2019) Railway track fastener defect detection based on image processing and deep learning techniques: a comparative study. Eng Appl Artif Intell 80:66–81
18. Chen K, Huang C, He J (2016) Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. High Voltage 1(1):25–33
19. Zhang WJ, Ge Q, Huang CY (2014) The research of photovoltaic array intelligent fault diagnosis based on the BP neural network. Adv Mater Res 936:2201–2206
20. Haba CG (2019) Monitoring solar panels using machine learning techniques. In: Modern power systems, pp 1–6
21. Li X, Li W, Yang Q (2019) Building an online defect detection system for large-scale photovoltaic plants. In: Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation. New York, pp 253–262
22. Greco A, Pironti C, Saggese A, Vento M, Vigilante V (2020) A deep learning based approach for detecting panels in photovoltaic plants. In: Proceedings of the ACM International conference proceeding series. Las Palmas de Gran Canaria, Spain
23. Massimiliano DB, Leonardi F, Messina F, Santoro C, Vasilakos A (2018) Anomaly detection and predictive maintenance for photovoltaic systems. Neuro Computing 3108:59–68
24. Lva F, Wenb C, Baao Z, Liua M (2016) Fault diagnosis based on deep learning. In: 2016 American control conference (ACC), pp 6851–6856
25. Li X, Yang Q, Lou Z, Yan W (2019) Deep learning based module defect analysis for largescale photovoltaic farms. IEEE Trans Energy Convers 34(1):520–529
26. Chine W, Mellit A, Lughy V, Malek A, Sulligoi G, Pavan AM (2016) A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks. Renew Energy 90:501–512
27. Lu X, Lin P, Chenga S, Lin Y, Chen Z, Wu L, Zhenga Q (2019) Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph. Energy Convers Manag 196:950–965

Covid Analysis Prediction Using Densenet Method in Deep Learning



M. Usha, P. Prittopaul, and D. Lekha

Abstract Coronavirus is a quickly spreading viral sickness that infects people, yet in addition creatures as well. Clinical research of COVID-19-tainted patients revealed that these individuals are frequently infected by a lung infection as a result of their interactions with this Corona Virus Disease. Chest CT-scans images are used for diagnosing lungs related problems. Deep learning is the best method of AI, which gives valuable examination to consider a lot of chest CT-Scan pictures that can fundamentally effect on screening of Covid-19. Image and statistical data were used for the evaluation of accuracy and mean value analysis. The DenseNet method is one of the Convolutional Neural Network methods, which achieves better performance during image pre-processing and prediction. The accuracy of our proposed system is up to 95 to 97%, respectively. This kind of system helps to analyze the COVID-19 infection in its early stages.

Keywords COVID-19 · Convolutional neural network · DenseNet · CT-scan

1 Introduction

Coronavirus is a serious illness issue where an enormous number of individuals lose their survives each day. In the previous decade, a few sorts of infections (like Flu, SARS [1], MERS, and so forth) came into the image [2, 3], anyway they address a few days or barely any months [4]. Numerous researchers are dealing with these sorts of infections, and not many of them are analyzed because of the accessibility of

M. Usha (✉) · P. Prittopaul
Velammal Engineering College, Chennai, India
e-mail: usha.m@velammal.edu.in

P. Prittopaul
e-mail: prittopaul.p@velammal.edu.in

D. Lekha
Department of CSE, RMK College of Engineering and Technology, Thiruvallur, Tamil Nadu,
India
e-mail: lekhacse@rmkcet.ac.in

immunizations arranged by them (i.e., Scientists or specialists). Covid-19 sickness is currently affecting the entire world [5], and the fundamental problem is that no single country's researchers have been able to develop an identical vaccine. The epic Covid illness started things out as throat contamination, and tainted patients of Covid-19 [6] should have been in confinement, perform legitimate screening, and hold sufficient assurance with anticipation to secure solid individuals. This contamination is a subsequent chain cycle [7] that exchanges starting with a single individual and then onto the next subsequent to interacting with Coronavirus tainted people. Emergency clinic staff, attendants, specialists, and clinical offices assume a fundamental part in the finding of this pandemic. A lot more methodologies have been pertained to lessen the effect of Covid-19. Clinical mirroring [8] is also a method for analyzing and anticipating the effects of Coronavirus on the human body. With the help of chest X-ray images [9, 10, 15] and CT images, healthy individuals and Covid-19 tainted patients can be separated in this way. X-ray images of solid and Coronavirus-infected individuals were transferred from numerous sources for use in the Covid-19 inquiry, and the DenseNet-121 model was assigned. The primary focus of this research is on using CNN models to order chest X-rays for Covid-affected patients. There are two major challenges that need to be resolved:

1. A CT-Scan image that it had been uploaded for process to envision COVID OR NOT-COVID that it doesn't belong to each the edges throughout the analysis therefore it ought to result as "INVALID DATA" or "INVALID IMAGE".
2. The input image that it's some totally different resolution, if the resolution is smaller than 64×64 throughout the compression or extension of image component there'll be some loss of knowledge are going to be occurring. So, consolidating the loss of knowledge throughout the compression and maintaining the accuracy.

2 Related Work

The authors of [11] used CNN models based on deep learning and compared their performance against three models: ResNeXt models, Xception, and Inception V3, as well as analyzing precisions. They focus on possible methods for classifying Covid-19-infected patients and make no claims to medical precision. In [12], a binary classification has generally been made to detect COVID-19. This study used convolutional neural networks and other deep learning architectures to diagnose COVID-19 from CXR pictures [13]. The methodologies used to train a Convolutional Neural Network with a dataset of more than 79, 500 X-Ray pictures gathered from various sources, including more than 8, 500 COVID-19 specimens, are illustrated by the authors in [14]. In [16], For the detection of coronavirus pneumonia-infected patients using chest X-ray radiographs, five pre-trained convolutional neural network-based models (InceptionV3, Inception-ResNetV2, ResNet50, and ResNet101, ResNet152) have been proposed [22]. Iohannis et al. [17] analyze the performance of state-of-the-art convolutional neural network architectures [21]. Authors in [18] hypothesized that

AI's deep learning algorithms could be able to remove COVID-19's unique visual characteristics and provide a clinical diagnosis before the pathogenic test, saving vital time for disease control. The photos were classified by the authors in [19, 20] using a Convolutional Neural Network (CNN), Softmax, Support Vector Machine (SVM), Random Forest, and K closest Neighbour AI methods (KNN). In [23], the authors examine how to use a Convolutional Neural Network (CNN)-based algorithm to a chest X-ray dataset in order to identify pneumonia.

3 Methodologies

3.1 DenseNet

DenseNet is one of the most recent neural network discoveries for visual item recognition. For certain core contrasts, it's pretty similar to ResNet. ResNet utilizes an added substance strategy (+) that consolidates the past layer (character) with the future layer, though DenseNet links (.) the yield of the past layer with the future layer.

The DenseNet is divided into DenseBlocks, each of which has many filters with similar measurements. DenseNet begins with a convolution layer and a pooling layer, then moves on to a dense block, Transition Layer, and lastly a dense block, followed by a classification layer. The DenseNet was illustrated in Fig. 1 [24, 25]. The following is the sequencing of each convolutional block: BatchNormalization comes first, then ReLU activation, and finally the Conv2D layer.

4 Proposed Work

COVID analysis can be done by CT-Scan images of different people. The image, which it received as input from the user, was compressed to 64x64 size in order to maintain the grayscale level. Pre-trained models are used for training the images

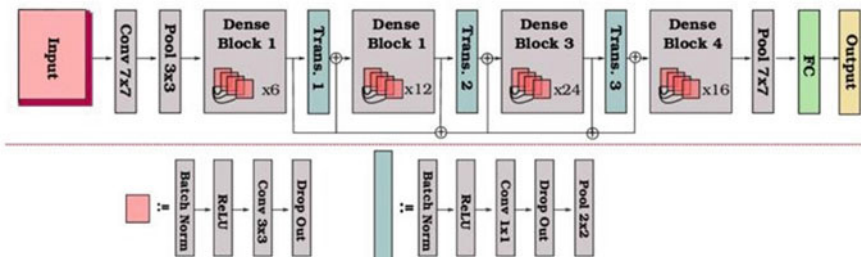


Fig. 1 Dense blocks and layers

of COVID and Normal types. The model used for the classification of images is DenseNet-121. DenseNet-121 is the CNN technique or a method that contains 121 layers to pre-process the image in the dataset. The images in the dataset were processed one by one through hidden layers of the DenseNet. After passing the output layer, a mean value and variance are obtained from the trained images. Once the values are received, they need to be data augmented. Data augmentation is a technique for increasing the size of a prepared dataset by modifying images in the dataset in various ways. The input image will be processed with the existing dataset by a model algorithm and then it results in an array that contains possibility values of COVID and Normal, based on which has maximum value will be decided either of cases and sends the output to the server.

4.1 Need for Covid Detection

- Building a COVID detection system helps to find out whether a person has COVID or not.
- It helps to find out COVID in the very early stages itself.
- It gives the results like what is the possibility a person can get Corona Virus Disease; based on that prediction, precautions are made for safety.
- If the COVID is evolved in the near future, by using image samples like CT Scans, Chest X-ray images, we can able to analyze the corona virus.
- It is so robust and availability.

4.2 Dataset

A publicly available SARS-CoV-2 CT scan dataset, comprising 1252 CT scans that are positive for SARS-CoV-2 infection and 1230 CT scans for patients convinced by 2482 CT scans, SARS-CoV-2, altogether. These statistics were gathered from real patients at emergency rooms in Sao Paulo, Brazil. The goal of this dataset is to encourage the development of AI systems that can determine whether a person is infected with SARS-CoV-2 by looking at his or her CT scans. This dataset is provided to the model in order for it to develop and produce results.

4.3 Server Creation

A localhost server is created, which interacts with the user to get the input image. The server act as a front end of the system, which gets the input and displays the output to the user.

4.4 Identification of Covid

After receiving the image, the image will be resized to 64×64 and converted into an array. The collection of images from the SARS-CoV-2 dataset, which contains the images of COVID and NORMAL, will be trained by using DN (DenseNet) method and performing all the performance, accuracy, and loss operations. This is called Transfer learning, which means obtaining a result from trained images. During the prediction, the input image will be compared with all the other images through the HL (Hidden-Layers) or FCL (Fully Connected Layer). Finally, the result will be moved to OL (Output-Layer) or DL (Dense-Layer) which helps to get an array that has a value of possibilities of COVID and NORMAL from the image.

4.5 System Architecture

1. The web server run as a back-end which allows the user to get the input image from the user as depicted in Fig. 2.

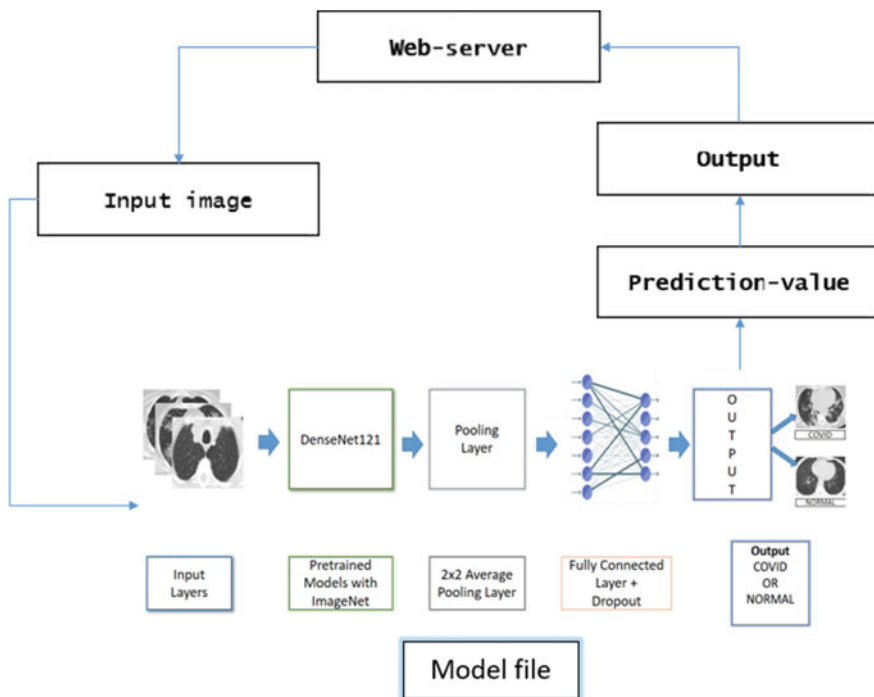
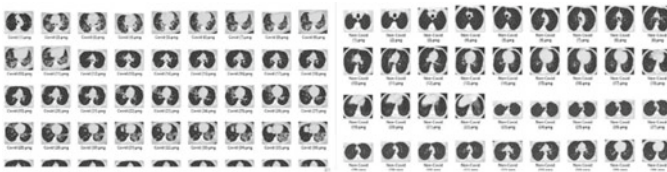


Fig. 2 Flow of Entire process

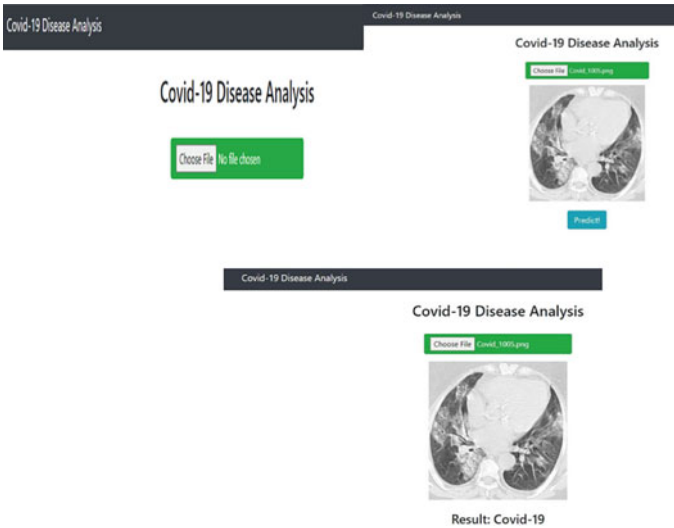
2. The image is sent through the DenseNet Algorithm and hidden layers in order to obtain the mean value and variance. These values will be compared with the pre-trained models from the dataset.
3. As the probabilities are received, it contains the possibilities of COVID and Normal values which have more value than other the output will be resulted in the server or displayed.

5 Implementation and Results

The result was an array that contains the values of possibilities of COVID and NORMAL. The output will be based on which class value will be more than another will; it is like either COVID or NORMAL. The output will be displayed on a web server.



Data Set



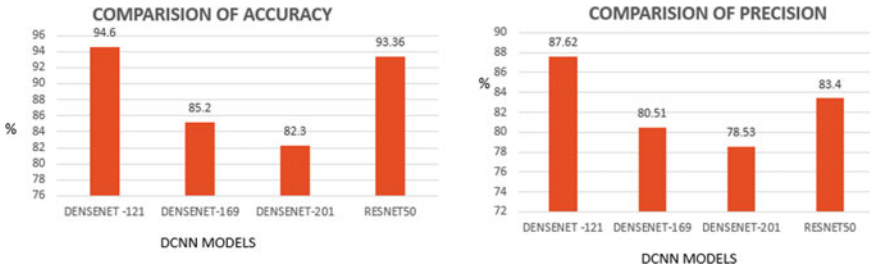


Fig. 3 Comparison of DCNN models based on accuracy

Figure 3 shows the performance analysis of accuracy and precision with ResNet50, DenseNet-121, DenseNet-169, and DenseNet-201. DenseNet-121 Provides better results across competitive datasets, as compared to their standard CNN.

6 Conclusion and Future Enhancement

DenseNet is one of CNN’s deep learning techniques which is more efficient and provides accurate results. This strategy is secure adequate, reliable, and accessible to use. There is no demand for specialized hardware to install this system. It was constructed by obtaining medical image datasets and a local host server.

In the future, a large dataset for Lung CT-Scan images can be considered to validate this model to achieve better accuracy. Further, the system will be developed for the detection of other diseases like pneumonia, influenza, and lung cancer so that it acts as a single platform that gives the results for the other diseases too.

References

- Ooi GC, Khong PL, Müller NL, Yiu WC, Zhou LJ, Ho JC, Lam B, Nicolaou S, Tsang KW (2004) Severe acute respiratory syndrome: temporal lung changes at thin-section CT in 30 patients. *Radiology* 230(3):836–844
- Wong KT, Antonio GE, Hui DS, Lee N, Yuen EH, Wu A, Leung CB, Rainer TH, Cameron P, Chung SS, Sung JJ (2003) Severe acute respiratory syndrome 228(2):401–406
- Xie X, Li X, Wan S, Gong Y (2006) Mining x-ray images of SARS patients. In: *Data mining*. Springer, Berlin, pp 282–294
- Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Zhang L, Fan G, Xu J, Gu X, Cheng Z (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* 395(10223):497–506
- Wang Y, Hu M, Li Q, Zhang XP, Zhai G, Yao N (2020) Abnormal respiratory patterns classifier may contribute to large-scale screening of people infected with COVID-19 in an accurate and unobtrusive manner. arXiv preprint [arXiv:2002.05534](https://arxiv.org/abs/2002.05534)

6. Roosa K, Lee Y, Luo R, Kirpich A, Rothenberg R, Hyman JM, Yan P, Chowell G (2020) Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th. *Infect Dis Model* 5:256–263
7. Yan L, Zhang HT, Xiao Y, Wang M, Sun C, Liang J, Li S, Zhang M, Guo Y, Xiao Y, Tang X (2020) Prediction of criticality in patients with severe Covid-19 infection using three clinical features: a machine learning-based prognostic model with clinical data in Wuhan. *medRxiv*
8. Xu B, Meng X (2020) A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)
9. Ajlan AM, Ahyad RA, Jamjoom LG, Alharthy A, Madani TA (2014) Middle East respiratory syndrome coronavirus (MERS-CoV) infection: chest CT findings. *Am J Roentgenol* 203(4):782–787
10. Kanne JP (2020) Chest CT findings in 2019 novel coronavirus (2019-nCoV) infections from Wuhan, China: key points for the radiologist
11. Gupta M, Taneja S, Jude Hemanth D (2021) Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl Intell* 51:1690–1700
12. Sevi M, Aydin I (2020) COVID-19 detection using deep learning methods. In: International conference on data analytics for business and industry: way towards a sustainable economy (ICDABI). IEEE
13. Alghamdi HS, Amoudi G, Elhag S, Saeedi K, Nasser J (2021) Deep learning approaches for detecting COVID-19 from chest X-Ray images: a survey. *Expert Syst Appl* 164
14. Arias-Londoño JD, Gómez-García JA, Moro-Velázquez L, Godino-Llorente JI (2020) Artificial intelligence applied to chest X-Ray images for the automatic detection of COVID-19. A Thoughtful Evaluation. *arXiv:2011.14259*
15. Shorten C, Khoshgoftaar TM, Furht B (2021) Deep learning applications for COVID-19. *J Big Data* 8(18). <https://doi.org/10.1186/s40537-020-00392-9>
16. Narin A, Kaya C, Pamuk Z (2020) Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*
17. Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilising transfer learning with convolutional neural networks. *Phys Eng Sci Med* 1
18. Meng BX, A deep learning algorithm using CT images to screen for corona virus disease (COVID-19)
19. Alqudah AM, Qazan S, Alqudah A (2020) Automated systems for detection of COVID-19 using chest X-ray images and lightweight convolutional neural networks
20. Iyas M, Rehman H, Nait-ali A (2020) Detection of Covid-19 from chest X-ray images using artificial intelligence: an early review. *arXiv preprint arXiv:2004.05436*
21. Maghdid HS, Asaad AT, Ghafoor KZ, Sadiq AS, Khan MK (2020) Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms. *arXiv preprint arXiv:2004.00038*
22. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Acharya UR (2020) Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Comput Biol Med*, 103792
23. Yadav SS, Jadhav SM (2020) Deep convolutional beam network based medical image classification for disease diagnosis. *J Big Data* 6(1):113
24. <https://towardsdatascience.com/understanding-and-visualizing-densenets-f688092391a>
25. Huang G, Liu Z, van der Maaten L (2018) Densely connected convolutional networks

Feature Extraction Based on GLCM and GLRM Methods on COVID-19 Dataset



N. Suganthi and K. Sarojini

Abstract COVID-19 epidemic had devastating effects on both the economic and social infrastructures of all countries in the world. Several researches are still being carried out in order to develop effective models for the diagnosis and treatment of COVID-19 patients. A COVID-19 infected person may experience dry cough, muscle ache, brain pain, fever, sore throat, and mild to severe respiratory illness. At the same time, it has a negative impact on the lungs. The severity of COVID-19 contamination over the lungs can be examined using X-Ray and CT scan images of the chest. The examination of the severity of disease is carried out by Manual characterization. However, it may lead to human error. To overcome this drawback, an exact and proficient indicative tool is highly required. Hence, this research provides a new topology for COVID-19 diagnosis using CT scan images. In this topology, features from the images are extracted using Gray Level Co-event Matrix (GLCM), Gray Level Run Length Matrix (GLRM). An automatic classification is carried out with supervised ML algorithms. In order to examine the effectiveness of the proposed model, an experiment was carried out on the COVID-19 Dataset. Various performance evaluation metrics are utilized to identify the best ML method.

Keywords COVID-19 · SVM · NB · DT · KNN

1 Introduction

COVID-19 is a widespread dangerous infectious disease, which is caused by severe respiratory syndrome. By December 2019, a coronavirus outbreak had been discovered in Wuhan, China. The outbreak of this COVID-19 has posed serious health and economic risks around the world. Fever, cough, muscle soreness, breathing problem,

N. Suganthi (✉) · K. Sarojini
Department of Computer Science, LRG Govt Arts College For Women, Tirupur, India
e-mail: suganca@gmail.com

K. Sarojini
e-mail: saromaran@gmail.com

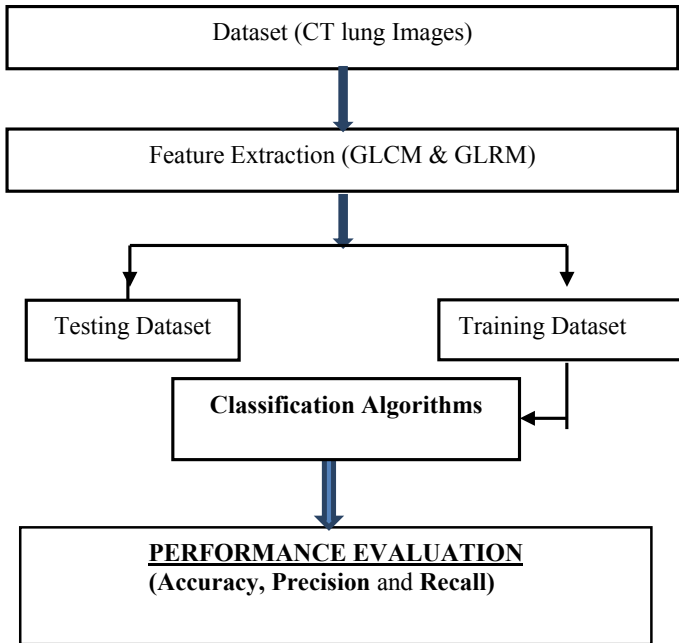


Fig. 1 Feature extraction

diarrhea, etc., are some of the common symptoms of COVID-19 [1]. Computer-aided diagnostic (CAD) systems have been formulated to assist clinicians in making quick diagnoses of COVID-19 using CT scan. To categorize Infected/Not Infected images, these systems typically utilize various processing stages such as preprocessing, extraction, and classification. Several topologies with standard ML algorithms have been proposed so far for detecting CT scan images. The capacity of pre-trained ML models and multiple versions of Covid-19 were explained in this research (Fig. 1).

2 Literature Review

The impact of contour wave transformation on gray correlation degree and noise intensity of various medical images is investigated by Li et al. [2]. In this, the Bayesian threshold is improved. Multiple thresholds confine the contour features of medical pictures and correlation properties of contour wave coefficients improved the middle threshold function. The wavelet MRA approach was implanted to lower the dimension of the medical image, and the matching threshold search space was obtained. In order to discover the optimal quasi-threshold in the search space, a GA was applied. A histogram of the medical image was established using this value. The optimal

feature extraction threshold of the medical image was obtained using the golden section approach. The experimental findings show that the suggested approach can extract contour feature information from an image quickly, with an optimum feature extraction effect and high feature extraction efficiency [2].

Chander et al. [3] proposed a new technique for detecting and classifying brain tumors. The characteristics were extracted from segmented medical images using the GLCM approach. They presented the system and implemented an SVM classifier for categorizing benign and malignant tumors, with the proposed system achieving a classification accuracy of 93% [3].

Deepika et al. [4] proposed a novel classifier model. In this, MRI data is converted into grayscale images using the T2-weighted preprocessing approach. Using the nearby twofold (LBP) technique, features are detected from the preprocessed images. Uncorrelated items are removed using PCA topology. SVM classifier is utilized to determine whether an MR image is normal (benign) or abnormal (malignant) [4].

The performance of three distinct feature extraction approaches for the classification of normal and abnormal patterns in mammograms was evaluated and compared by Nithya and Santhi [5]. This work utilized intensity histogram, GLCM (and intensity-based features feature extraction approaches. ANN-based supervised classifier system is used. According to the findings of the experiments, the GLCM approach outperformed the other two methods.

SVM and KNN classifiers, with GLDM and Gabor feature extraction algorithms, have been proposed by Neeta and Mahadik [6] for Mammography. Experiments were carried out using the MIAS database. The results suggest that combining the GLDM feature extractor with the SVM classifier produces better results than the other two approaches.

3 Proposed Methodology

3.1 Feature Extraction

The process of assessing image texture involves feature extraction. The findings provide a clearer picture of how texture and object shape are determined. For better handling, the algorithm should be switched to a smaller dimension when it has a larger input data set. Feature extraction is the process of converting an input image into a standard set of features. The grouped pixels were converted to numerical data by using the feature extraction procedure on CT images in the normal and COVID-19 categories. GLCM for extracting statistical features and GLRM for extracting run length characteristics are considered in this study work [7]. For normal and COVID-19-positive images, the features derived by GLCM and GLRM were listed here (Table 1).

For every image (both normal and affected images), a total of 29 characteristics (22 GLCM Features + 7 GLRM Features) were retrieved. The intensity characters are

Table 1 Overall features

Methods	Features
GLCM based features	Auto correction, contrast, correlation, cluster prominence, cluster, shade dissimilarity, energy, entropy, homogeneity, maximum probability, sum of squares (Variance), sum average, sum variance, sum entropy, difference variance, difference entropy, information measures of correlation1, information measures of correlation2, inverse difference, inverse difference normalized, inverse difference moment normalized
GLRM based features	SRE, LRE, RLU, GLN, RP, LGLRE and HGLRE

determined by the extracted features, the spatial dependencies of the gray levels on different angles are determined by GLCM, and the image coarse characteristics are determined by the run length. By providing higher level statistical information, the GLRM considerably increases the intensity of the class area. As a result, the recovered features provide more information on image intensity, shape, texture, and location. Normal cells are smaller in size and have a regular shape, however, abnormal cells are dispersed and large in size. The data set created by these two feature extraction procedures is fed into classifiers as input.

3.2 Gray Level Co-Occurrence Matrixes

A GLCM is a framework in which the number of lines and sections in an image of a surface is equivalent to the number of distinct dim levels or pixel values in that picture. The GLCM's substance is used in the texture including computations to give a proportion of the power variation at the pixel of interest. The co-event framework is often processed based on two boundaries: the general distance between the pixel pair d , as measured in pixel number, and their relative.

Direction. On a regular basis, it is quantified in four directions (e.g., 0° , 45° , 90° , and 135°).

The figure beneath addresses the arrangement of the GLCM of the dark level (4 levels) picture at the distance $d = 1$ and the heading of 0° (Figs. 2 and 3).

3.3 Gray Level Run Length Matrix

GLRM is the quantity of runs with pixels of dark level I and run length j for the provided guidance. A bunch of back to back pixels with a similar dark level is known as a dim level run. The quantity of pixels in a run is the run length. GLRM likewise has four counterbalances like GLCM (Figs. 4 and 5).

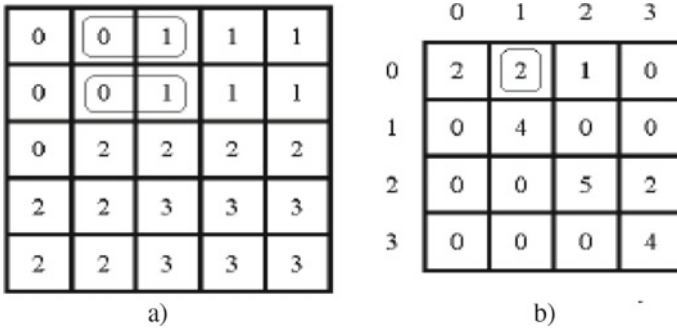


Fig. 2 Arrangement of the GLCM. **a.** An image with 4 Gy level images, **b.** GLCM for distance 1 and direction 0°

Fig. 3 Direction of GLCM age. From the middle (0) to the pixel 1 addressing course = 0° with distance d = 1, to the pixel 2 course = 45° with distance d = 1, to the pixel 3 course = 90° with distance d = 1, and to the pixel 4 course = 135° with distance d = 1

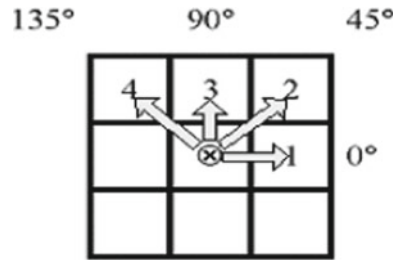


Fig. 4 Matrix of Image 4X4 pixels

1	2	3	4
1	3	4	4
3	2	2	2
4	1	4	1

Fig. 5 GLRM matrix

Gray Level	Run Length (j)			
i	1	2	3	4
1	4	0	0	0
2	1	0	1	0
3	3	0	0	0
4	3	1	0	0

4 Classification Techniques

In this research, the classification problem has been solved using supervised ML algorithms like normal classification problem which uses scientific features to solve it. Here, one of the SVM is employed to classify COVID-19 disease. The learning instances in the training data are used to aid the learning models to train the supervised learning to make it proficient. As a result, they can attain a high level of categorization accuracy. This research applies some ML classification algorithms such as:

- Decision tree (DT) Algorithm [6]
- K-Nearest Neighbor (KNN) [7]
- Support Vector Machines (SVM) [8]
- Naive Bayes (NB) [9]

And also demonstrates all classification algorithms' performance on extracted features.

5 Results and Discussion

The intended system is executed in MATLAB2018a with ML packages. The intended technique has been tested with the datasets after Feature extraction of the CT scan images data ser. Initially, 2484 images were taken for analysis in this 1341 are normal images and 1143 images are COVID-19 affected images. After data collection, feature extraction techniques are applied to extract 29 appropriate features from raw data. Numerous evaluation processes are considered to be the most important phase, since the proposed algorithm performs with minimum feature vectors from the feature extraction and performance of classification model is calculated. The effort of the proposed methodology has been assessed using feature selection and classification algorithm provided by the MATLAB machine learning tool and classifies the normal and COVID-19 affected images which are presented in the extracted dataset. Assessing all the results achieved by all the employed algorithms, the experimental results proved that the SVM classification algorithm has increased the performance of the algorithm.

5.1 Performance Measures Parameters

The dataset's characteristics are employed in the training and testing processes. 80% percent of the data was utilized for training and 20% was used for testing in the assessment. The following parameters are used to evaluate the different algorithms: TP, FP, TN, FN, Precision, Recall, F1-Score, and classification Accuracy [10].

- True Positive—If the normal feature is correctly identified, then it is TP
- True Negative—If the abnormal feature is correctly classified as abnormal, then it is TN
- False Positive—If the normal feature is wrongly identified, then it is FP
- False Negative—If the abnormal feature is wrongly classified as normal feature, then it is FN
- Accuracy value is the proportion of the number of accurate predictions. It can be calculated using the below equation

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

- Precision is the proportion of anticipated positive models which truly are positive

$$Precision = \frac{TP}{TP + FN}$$

- Recall, also known as hit rate or affectability, is a metric that measures how well a classifier can recognize positive models.

$$Recall = \frac{TP}{TP + FN}$$

A confusion matrix is used in this research to describe the performance of the classification algorithm. It also permits easy recognition of uncertainty between classes. The goal of this procedure is to evaluate the effectiveness of various classification methods. The categorization results for the extracted datasets are shown in the table below. Different cases have been used to evaluate it (Fig. 6 and Table 2).

The characteristics such as Accuracy, Precision, and Recall have been determined to evaluate the performance, and it can also be compared to current algorithms, demonstrating that the suggested FS technique beats the others. The suggested work demonstrates that the SVM technique has greater classification accuracy.

6 Conclusion

The features are extracted for the classification of micro-calcifications. In this work, two types of texture description methods such as GLCM and GLRM are analyzed over the Covid-19 image. It also focused on the classification of the COVID-19 dataset as either positive or negative. From the results, it became clear that a higher classification accuracy of 77.96% is achieved for COVID-19 and non-COVID-19 classification using the SVM method. From the experimental output, it is interpreted

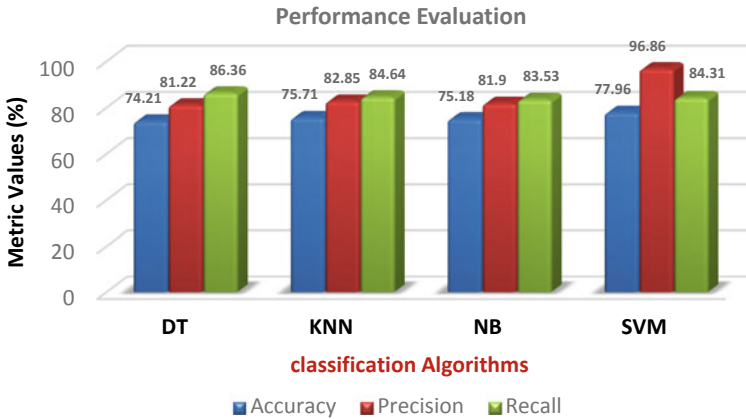


Fig. 6 Performance evaluation

Table 2 Classification datasets

classification Algorithms	DT	KNN	NB	SVM
Accuracy	74.21	75.71	75.18	77.96
Precision	81.22	82.85	81.90	96.86
Recall	86.36	84.64	83.53	84.31

that, SVM achieved better accuracy when compared with other DT, KNN, and NB algorithm results. Hence, it is concluded that, the SVM method is the best ML method for the analysis of COVID-19.

References

1. Elmuogy S, Hikal NA, Hassan E (2021) An efficient technique for CT scan images classification of COVID-19. 5225–5238
2. Li W, Huang Q, Srivastava G (2021) Contour feature extraction of medical image based on multi-threshold optimization. *Mobile Netw Appl Springer* 26:381–389
3. Chander PS, Soundarya J, Priyadharsini R (2019) Brain tumour detection and classification using K-means clustering and SVM classifier. In: Abdul Majeed PP, Mat-Jizat J, Hassan M, Taha Z, Choi H, Kim J (eds) *Lecture notes in mechanical engineering RITA 2018*. Springer, Singapore, pp 49–63
4. Deepika K, Bodapati JD, Srihitha RK (2019) An efficient automatic brain tumor classification using LBP features and SVM-based classifier. In: *International conference on computational intelligence and data engineering*, vol. 28. Springer, Singapore
5. Nithya R, Santhi B (2011) Comparative study on feature extraction method for breast cancer classification. *J Theoret Appl Inform Technol* 33(2)
6. Neeta VJ, Mahadik SR (2015) Implementation of segmentation and classification techniques for mammogram images. *Int J Innov Res Sci Eng Technol* 4(2)

7. Singh R (2018). A comparison of gray-level run length matrix and gray-level co-occurrence matrix towards cereal grain classification
8. Barbareschi M, Barone S, Mazzocca N (2021) Advancing synthesis of decision tree-based multiple classifier systems: an approximate computing case study. *Springer KnowlInfSyst* 63:1577–1596
9. Dang Y, Jiang N, Hu H et al (2018) Image classification based on quantum K-Nearest-neighbor algorithm. *Springer Quantum Inf Proc* 17:239
10. Manju S, Helenprabha K (2021) A structured support vector machine for hyperspectral satellite image segmentation and classification based on modified swarm optimization approach. *J Ambient Intell Human Comput* 12:3659–3668
11. Khatami A, Araghi S, Babaei T (2019) Evaluating the performance of different classification methods on medical X-ray images. *SN Appl Sci* 1:1154
12. Ghori KM, Imran M, Nawaz A et al (2020) Performance analysis of machine learning classifiers for non-technical loss detection. Springer, *J Ambient Intell Human Comput*
13. Amadasun M, King R (1989) Textural features corresponding to textural properties. *IEEE Trans Syst Man Cybernet* 19(5):1264–1274
14. Haralick RM, Shanmugan K, Dinstein I (1973) Textural features for image classification', *IEEE Trans Syst Man Cybernet SMC-3(6):610–621*
15. Haralick RM (1979) Statistical and structural approaches to texture. In: *Proceedings of the 4th International Joint Conference Pattern Recognition*. pp 45–60
16. Gulsrud TO (2000) Texture analysis of digital mammograms. PhD Thesis, Aalborg University, Stavanger, USA, pp 30–32
17. Halpern EJ (2006) Contrast-enhanced ultrasound imaging of prostate cancer. *Rev Urol* 8(Suppl 1):S29

Memory Augmented Distributed Monte Carlo Tree Search Algorithm-Based Content Popularity Aware Content Recommendation Using Content Centric Networks



A. P. Christopher Arokiaraj and D. Hari Prasad

Abstract The rapid advancements of social media networks have created the problem of overloaded information. As a result, the service providers push multiple redundant contents and advertisements to the users without adequate analysis of the user interests. The content recommendation without user interests reduces the probability of users reading them and the wastage rate of network load increases. This problem can be alleviated by providing accurate content recommendations with consideration of users' precise interests and content similarity. Content centric networking has been developed as the trending framework to satisfy these requirements and improve access to relevant information and reception by the desired user. The uses of message entity by giving a proper name, the users' real-time interests are identified and then the accurate and popular contents with high contextual similarity are recommended. An efficient content recommendation scheme is presented in this paper using Memory Augmented Distributed Monte Carlo Tree Search (MAD-MCTS) algorithm for ensuring minimum energy consumption in the CCN. The big data context of the users' social media data is considered in this study so that the complexity can be visualized and controlled to minimize the network complexities. Experiments are conducted on a benchmark as well as an offline collected Twitter dataset on Covid-19 and the results implied that the accuracy and convergence of the proposed MAD-MCTS outperform the other content recommendation algorithms.

Keywords Content centric networking · Content recommendation · Content push · Memory augmented distributed Monte Carlo tree search · Big social data · Social network · Recommender system · Covid-19 tweets

A. P. Christopher Arokiaraj (✉) · D. Hari Prasad
Sri Ramakrishna College of Arts and Science, Coimbatore, India
e-mail: christo4u2005@gmail.com

D. Hari Prasad
e-mail: dhp@srcas.com

Department of Computer Applications, Sri Ramakrishna College of Arts and Science,
Coimbatore, India

1 Introduction

Social media networks such as Facebook, Twitter, Instagram, YouTube, etc., have transformed the social interactions between humans into a gigantic space. The simplicity and cohesion in social media have led to an increasing amount of interactions. In addition to viewing and interacting with known members socially, social media networks provide a platform for communication between like-minded unknown people [1]. Based on the likes and interests of the users, the general content published on social networks are also recommended for the users. The major factors for these considerations are the geographical location, similarity of interests, popularity among other similar users, and contents from followers or friends. It provides access to all users to get information about all contents. Certain information is sensitive and is likely to reach all over the network and subsequently, data overload becomes the greatest issue to deal with for the networking access providers. They recommend a variety of informations including the information initiated by the users and popularizing it will lead to exceeding the users' capacity for browsing [2]. Thus, most of the recommended contents will be useless and the network load will waste significantly. It also negatively impacts the profits of the network service providers as their primary source of income through the advertisements will be reduced [3]. These problems have motivated several researchers to propose accurate content recommendation systems. Accurate content recommendation is often considered a multi-class classification problem in which the contents must be categorized based on different users' perceptive, interests, and requirements to achieve personalized recommendation. However, there are two complications in achieving these performances. They are (i) the precise description of the users' interests and (ii) the ever-increasing volume and diversity of the data associated with the users' interests. These two challenges have led to the development of out-of-the-box techniques [4]. CCN has been projected by Jacobson et al. [5] as a budding networking framework for analyzing the accurate information recommendation to satisfy the interests of legitimate users. It decides on what content to be shown to the users rather than where they need to be shown. This diversity improves the variety of contents for the users under their interests and also increases the probability of viewing them [6]. This new technology saves maximum network load and also reduces energy consumption than the other IP-based networking models. The concept of CCN also supports social media networks with great flexibility to retrieve and diffuse the contents to the users [7, 8]. CCN allows the service providers in providing accurate user perception and increases their revenue through adequate advertisements. This property can be exploited by designing an accurate content recommendation algorithm. Yet, the inclusion of big data is very challenging. Previous works on big data-based content recommendation are not adequate since the computational cost increases with the increased amount of data [9]. Additionally, the traditional models cannot support instant learning of the increasing big data which is considered to be a major drawback. Recently, context-based multi-armed bandit-based Montecarlo tree Search algorithm was presented

by Feng et al. [10]. This approach solved the problem of content recommendation from the perspective of big data through modeling of incessant resemblance among legitimate user's features. However, the limitations of MCTS reduce the overall effectiveness. The MCTS algorithm attempts to prune sequences that are less relevant and leads to loss of data. It also needs a huge number of iterations to effectively decide the most efficient direction of content similarity routing. As the tree growth becomes rapid, after a few iterations, it requires a huge amount of memory. Considering these limitations, an improved search algorithm is presented in this paper in the form of Memory Augmented Distributed Monte Carlo Tree Search (MAD-MCTS). MAD-MCTS uses the users' interests and popularity-based content density as the main features and models the accurate content recommendation problem as a stochastic multi-armed bandit problem. Then the MAD-MCTS algorithm is developed by enhancing the distributed property of MCTS with an effective Roulette selection and integrating the memory augmentation concept into the basic MCTS algorithm. Here, the distributed MCTS performs parallel processing through tree-splitting where the nodes are selected using Roulette wheel selection. Likewise, memory augmentation exploits generalization in online real-time search and reduces storage issues. Thus, the proposed approach can efficiently provide precise information recommendation in the big data setting with relatively less cost and also maintain the instant increase of data.

The rest of this article is structured as follows: The related study in Sect. 2. The system model, problem formulation, and the proposed MAD-MCTS-based content recommendation system is explained in Sect. 3. Performance results and comparative investigations are illustrated in Sect. 4. References of this research and possible future instructions are given in Sect. 5.

2 Related Works

Many studies have been presented in recent years on the topic of efficient content recommendation. Feng et al. [10] developed a distinguished context-based multi-armed bandit-based Monte Carlo tree search algorithm to solve the problem of precise push of the relevant information in the network. However, this approach has high energy consumption in the network model due to the wastage of recommended contents. Makki et al. [11] designed Twitter message recommendation based on user interests. It was tested on TREC 2015 Microblog track dataset and achieved 95% accuracy. However, this approach did not consider other features such as content location, popularity, etc. Kazai et al. [12] designed customized news and recommendations for blogging by using location-based filtering and profile of the legitimate users. It achieved 97% accuracy with reduced recommendation time. However, this method lacked the effectiveness to handle false positives.

Otsuka et al. [13] developed a hashtag recommendation system for Twitter data using TF-IDF method and MapReduce. It achieved 30% hashtag recall and 93%

accuracy. However, this method falsely identifies similar hashtags that are unrelated in real sense. Daptardar et al. [14] developed a personalized movie recommendation system using Twitter profile extraction and achieved 90% accuracy. Das et al. [15] also proposed a personalized movie recommendation system using Twitter data which used ranking methods. Although effective, this method has reduced the overall hit ratio. Mwinyi et al. [16] presented a predictive self-learning content recommendation system using short-term characterizations for multimedia contents. This approach provided accurate recommendations without decreasing the performance of the system in terms of response time and CPU utilization. However, it has high learning time and also has high false positives for advertisements since they are recommended based on benefits to providers.

Rutkowski et al. [17] developed a content-based recommendation system using a neuro-fuzzy approach that used the CUDA technology. This approach provided high accuracy of 97%, even before the back-propagation fine-tuning was done to the neural networks. However, this method has high complexity for low-end users. Yu et al. [18] designed a content-based goods image recommendation system using weighted feature indexing and binary tree. This approach was tested on a real goods image database with 98% accuracy. However, it has high memory requirements. Antony Rosewelt and Arokia Renjit [19] presented a content recommendation system using embedded feature selection and Fuzzy Decision Tree-based CNN. Although efficient, this approach has limited relevance in terms of recommended contents. Tahmasebi et al. [20] proposed the social movie recommender system from Twitter data using the deep auto-encoder network. This approach achieved 97% accuracy and a high hit ratio. However, the complexity of this method is higher than other methods. Motivated by the limitations of the methods in the literature, this study has presented an efficient and accurate content recommendation system using MAD-MCTS algorithm. It reduces energy efficiency and improves the accuracy of the content push. It also improves the profits for the service providers.

3 MAD-MCTS-Based Content Recommendation System

3.1 System Model and Problem Formulation

The content recommendation problem is modelled based on the general content flow structures. The user set is represented as \mathcal{U} and is divided into various user categories. The user set expands with increasing time and each user will be slotted in a time slot denoted as $t = 1, 2, 3, \dots, T$. The information is recommended when the user $u_i \in \mathcal{U}$ freshly opens or refreshes the home page of social media and sends a fascinating message to the server at t . Through the accelerated learning process, the service provider observes the message as two special kinds of interests, namely: Recommendation Request Message and Content request message. As the same user can log in

from multiple devices, the recommendations must be synchronized and updated regularly across all devices. The contents set is represented as $\mathcal{C} = \{c_1, c_2, \dots\}$ which will expand over time based on the uploading rate of content providers.

When the terminal accepts and acknowledges the Interest message, it reads the user’s context such as gender, location, education, recent interests, and relationships according to the message and determines the user type x_{at} by his/her context. The two main contexts, namely, friendship and following relationships are considered to form the content tree. Then the message type is identified. If it is CRM, the system replies to the content directly. If it is RRM, the system analyses the content tree of personalized data structure to select the content node with the highest estimated reward r_i of x_{at} and returns the content c_i .

The user set $\mathcal{U} = \{u_1, u_2, \dots\}$ has been modelled as a measurable dimensional space d_u to outline the users by their contexts. Each user $u_i \in \mathcal{U}$ is described by the d_u dimensional vector represented as u with slight abusing of notation. For simplicity of exposition and without loss of generality, $\mathcal{U} = [0, 1]^{d_u}$ is a unit hypercube. The dissimilarity between the two users $u_i, u_j \in \mathcal{U}$ has been defined as

$$\|u_i - u_j\| = \sqrt{\sum_{d=1}^{d_u} |u_i^d - u_j^d|^2} \tag{1}$$

Here, u_i^d and u_j^d are the d -th dimensions of u_i and u_j , respectively. The user space is divided into m_T subspace. Each subspace represents a user type, denoting a-th user type by x_a . The shape of the subspace is not restricted to illustrate the exact nature of users range. A larger range of users is represented by a rectangular subspace and a smaller range of users is represented by a smaller triangle. To balance the computing efficiency and personalized recommendation, $a \in [1, m_T]$ is set between the range.

For finding the similarity between the users’ interests, a content dissimilarity function $f(c_1, c_2)$ is used. The content space \mathcal{C} is equipped with a dissimilarity function $f: \mathcal{C}^2 \rightarrow [0, \infty]$ such that for all $c_1, c_2 \in \mathcal{C}$ and we get $f(c_1, c_2) \geq 0$ and $f(c_1, c_1) = 0$. This condition must be satisfied to provide an accurate content recommendation. The regret is the loss during the content recommendation. It is caused by a lack of knowledge of reward distribution. When defining one step regret at time t as $\Delta t = \mu_t^* - r_t$. Here μ_t^* is the optimal expected reward for its user type and r_t is the reward at the time t . The total expected regret can be modelled as

$$R_T = \mathbb{E} \left[\sum_{t=1}^T \Delta t \right] = \sum_{t=1}^T (\mu_t^* - \mathbb{E}(r_t)) \tag{2}$$

The goal of the proposed MAD-MCTS algorithm is to proficiently compute a recommendation strategy to minimize R_T . R_T illustrates the learning speed or the convergence rate to the best possible strategy. Thus, a sublinear regret-upper bound provides an effective way for the proposed algorithm and can converge to the optimal strategy.

3.2 Popularity Estimation

Every node counts locally the number of requests for each content name, and stores the pair (Content Name; Popularity Count) into a Popularity Table. Once a content name reaches locally a Popularity Threshold, the content name is tagged as being popular and if the node holds the content, it suggests its neighbour nodes cache it through a new Suggestion primitive. These suggestion messages are accepted or not based on local policies such as resource availability. As the popularity of content can decrease with time after the suggestion process, the Popularity Count is reinitialized according to a Reset Value to prevent flooding the same content to neighbours. This strategy influences directly the CCN node requirements. The popularity distribution model can be formulated using Zipf's law distribution [21]. According to this law, the popularity of content to be requested is directly proportional to the rank of content among all other contents. The popularity of content c_i is defined as with relation to its ranking order $Rank$. The value of popularity $Pop(c_i)$ is the popularity of i -th content, which is given as

$$Pop(c_i) \propto \frac{1}{Rank} \quad (3)$$

$$Pop(c_i) = \frac{\frac{1}{Rank}}{\sum_{k=1}^N \frac{1}{k}} \quad (4)$$

Here k is a random integer.

Using this formula, the universal and local popularity of the contents can be computed. The universal popularity $UP(c_i)$ is similar to the general popularity function.

$$UP(c_i) = \frac{\frac{1}{Rank}}{\sum_{k=1}^N \frac{1}{k}} \quad (5)$$

Similarly, local popularity can be computed by finding the expected popularity of Eq. (4). It will also make the hit ratio remain constant for a certain period of examination. The expected popularity is defined by an expected value of content requests made by the clients which are based on the popularity and probability of popularity. Therefore, if the content has a high value of the probability of popularity, then the expected number of request will be increased. The expected popularity of $\mathbb{E}(Pop(c_i))$ of each content c_i can be defined as

$$\mathbb{E}(Pop(c_i)) = \int_{t=1}^T Pop(c_i) \times Prob(Pop(c_i)) \quad (6)$$

Let $\phi : \mathcal{S} \rightarrow \mathbb{R}^D$ be a function to generate the feature representation of a state. For two states $s_1, s_2 \in \mathcal{S}$, the difference between $V^*(s_1)$ and $V^*(s_2)$ are approximated by a distance function $d(s_1, s_2)$ which is set to be the negative cosine of the two states' feature representations.

$$\varepsilon_{s_1, s_2} \approx d(s_1, s_2) = -\cos(\phi(s_1), \phi(s_2)) \quad (8)$$

To keep the inner product unbiased, a feature hashing function h is introduced such that the feature representation becomes $\phi(s_1) = h(\zeta(s_1))$ and ζ is the normalized state function. The expected value becomes

$$\mathbb{E}[\cos(\phi(s_1), \phi(s_2))] = \cos(\zeta(s_1), \zeta(s_2)) \quad (9)$$

δ_{s_2} is the term corresponding to the sampling error, which is inversely proportional to the simulation numbers: $\delta_{s_2} \propto 1/N_{s_2}$ where N_{s_2} is the number of nodes in the state s_2 . By combining Eq. (8) with the concept of e^y is very close to $y + 1$ for any small value of y , the approximation of w is possible.

$$w_i = \frac{N_i \exp(-\frac{d(i, s_2)}{\tau})}{\sum_{j=1}^M N_j \exp(-\frac{d(j, s_2)}{\tau})} \quad (10)$$

By applying these approximations to the MCTS, the MAD-MCTS can be formed as a kernel-based method with $\frac{\exp(-\frac{d(i, s_2)}{\tau})}{\sum_{j=1}^M \exp(-\frac{d(j, s_2)}{\tau})}$ becoming the kernel function and τ becomes the smoothing factor.

One memory M is maintained in the proposed approach. Each entry of M corresponds to one particular state $s_1 \in \mathcal{S}$. It contains the state's feature representation $\phi(s_1)$ as well as its simulation statistics $V(s_1)$ and $N(s_1)$. There are three operations to access M : update, add and query.

Update: If the simulation statistics of a state s_1 have been updated during MCTS, we also update its corresponding values $V(s_1)$ and $N(s_1)$ in the memory.

Add: To include state s_1 , we add a new memory entry $\{\phi(s_1), V(s_1), N(s_1)\}$. If s has already been stored in the memory, we only update $V(s_1)$ and $N(s_1)$ at the corresponding entry. If the maximum size of the memory is reached, we replace the least recently queried or updated memory entry with the new one.

Query: The query operation computes a memory-based approximate value given a state $s_2 \in \mathcal{S}$. We first find the top M similar states in M based on the distance function $d(\cdot, s_2)$. The approximated memory value is then computed by $V_M(s_2) = \sum_{i=1}^M w_i(s_2) \cdot V(i)$ where the weights are calculated using Eq. (9).

4 Experiments and Results

The evaluation of the proposed MAD-MCTS-based content recommendation is performed using MATLAB R2016b under a controlled environment. Experiments are performed to compare the efficiency of MAD-MCTS on the benchmark as well as an offline Covid-19 dataset. The performance ratio considered in this study is accuracy, precision, recall, hit ratio, and push time. The existing MAB-MCTS is also implemented for comparison.

4.1 Benchmark Dataset

The benchmark dataset used is the Health News in Twitter Dataset available at UCI repository (<https://archive.ics.uci.edu/ml/datasets/Health+News+in+Twitter>). It was gathered by Karami et al. [23] for their research study. The dataset consists of 58,000 instances and 25,000 attributes. It contains health news from 15 major health news agencies such as BBC, CNN, and NYT. The results obtained for the benchmark data files of BBC, CNN, and Everyday news are shown in Table 1.

From Table 1, it can be seen that the proposed MAD-MCTS has achieved better performance results. The performance on all three data files has been significant and the push times are also considerably lesser.

4.2 Covid-19 Dataset

The dataset consists of tweets collected from multiple topics of interest related to Covid-19. The twitter datasets are collected using the Twitter streaming API. A total of 242,623 tweets were collected without duplication and after pre-processing and normalization 17,000 tweets are used in the final dataset. Table 2 shows the results obtained for the Covid-19 dataset for both the proposed MAD-MCTS and the existing MAB-MCTS.

Table 1 Results for benchmark dataset

Metrics/data files	Existing MAB-MCTS			Proposed MAD-MCTS		
	BBC	CNN	Everyday	BBC	CNN	Everyday
Accuracy (%)	91.44	93.24	92.02	91.44	93.24	92.02
Precision (%)	87.14	90.0	82.5	87.14	90.0	82.5
Recall (%)	86.67	85.38	83.33	86.67	85.38	83.33
Hit ratio	0.82	0.825	0.9643	0.82	0.825	0.9643
Push time (seconds)	18.6427	15.3770	17.0191	18.6427	15.3770	17.0191

Table 2 Results for Covid-19 dataset

Metrics/data files	Existing MAB-MCTS	Proposed MAD-MCTS
Accuracy (%)	84.33	92.16
Precision (%)	87.98	92.93
Recall (%)	91.65	98.92
Hit ratio	0.75	0.8667
Push time (seconds)	35.667	21.0954

From Table 2, it can be seen that the proposed MAD-MCTS has achieved better performance results even for the Covid-19 dataset. MAD-MCTS has high values of all parameters. It has 7.83% high accuracy, 4.96% high precision, 7.27% higher recall, 11.67% higher hit ratio, and 14.6 s lesser push time than the existing MAB-MCTS.

5 Conclusion

Accurate content recommendation or content push is a highly important research topic. This paper presented an improved content recommendation system using MAD-MCTS algorithm. This approach included the use of the popularity of the content to reduce the unwanted contents being recommended. Then the contents are recommended effectively using the proposed approach with the memory augmented search process to reduce the slow training and improve accuracy. The results indicated that the proposed MAD-MCTS-based approach is highly efficient than the existing MCTS model. In future, the content freshness will be investigated to improve user satisfaction and maintain the revenue for service providers.

References

1. Greenwood S, Perrin A, Duggan M (2016) Social media update 2016. *Pew Res Center* 11(2):1–18
2. Lai LS, To WM (2015) Content analysis of social media: a grounded theory approach. *J Electron Commer Res* 16(2):138
3. Lee J, Hong IB (2016) Predicting positive user responses to social media advertising: the roles of emotional appeal, informativeness, and creativity. *Int J Inf Manage* 36(3):360–373
4. Lou L, Koh J (2018) Social media advertising effectiveness: a conceptual framework and empirical validation. *Asia Pac J Inform Syst* 28(3):183–203
5. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking named content. In: *Proceedings of the 5th international conference on emerging networking experiments and technologies*, pp 1–12
6. Xu Y, Li Y, Lin T, Wang Z, Niu W, Tang H, Ci S (2014) A novel cache size optimization scheme based on manifold learning in content-centric networking. *J Netw Comput Appl* 37:273–281
7. Li C, Liu W, Wang L, Li M, Okamura K (2015) Energy-efficient quality of service-aware forwarding scheme for content-centric networking. *J Netw Comput Appl* 58:241–254

8. Kumar S, Tiwari R (2020) Optimized content centric networking for future internet: dynamic popularity window based caching scheme. *Comput Netw* 179:107434
9. Bhuiyan MZA, Wang G, Tian W, Rahman MA, Wu J (2017) Content-centric event-insensitive big data reduction in internet of things. In: *GLOBECOM 2017–2017 IEEE global communications conference*. IEEE, pp 1–6
10. Feng Y, Zhou P, Wu D, Hu Y (2018) Accurate content push for content-centric social networks: a big data support online learning approach. *IEEE Trans Emerg Top Comput Intell* 2(6):426–438
11. Makki R, Soto AJ, Brooks S, Milios EE (2016, August) Twitter message recommendation based on user interest profiles. In: 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE, pp 406–410
12. Kazai G, Yusof I, Clarke D (2016) Personalised news and blog recommendations based on user location, Facebook and Twitter user profiling. In: *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval*, pp 1129–1132
13. Otsuka E, Wallace SA, Chiu D (2016) A hashtag recommendation system for twitter data streams. *Comput Soc Netw* 3(1):3
14. Daptardar SS, Apte S, Pednekar P (2018) Personalized movie recommendation system using twitter profile extraction. *Int J Eng Comput Sci* 7(10):24366–24371
15. Das D, Chidananda HT, Sahoo L (2018) Personalized movie recommendation system using twitter data. In: *Progress in computing, analytics and networking*. Springer, Singapore, pp 339–347
16. Mwinyi IH, Narman HS, Fang KC, Yoo WS (2018) Predictive self-learning content recommendation system for multimedia contents. In: 2018 wireless telecommunications symposium (WTS). IEEE, pp 1–6
17. Rutkowski T, Romanowski J, Woldan P, Staszewski P, Nielek R, Rutkowski L (2018) A content-based recommendation system using neuro-fuzzy approach. In: 2018 IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE, pp 1–8
18. Yu L, Han F, Huang S, Luo Y (2018) A content-based goods image recommendation system. *Multimedia Tools Appl* 77(4):4155–4169
19. Antony Rosewelt L, Arokia Renjit J (2020) A content recommendation system for effective e-learning using embedded feature selection and fuzzy DT based CNN. *J Intell Fuzzy Syst* (Preprint):1–14
20. Tahmasebi H, Ravanmehr R, Mohamadrezaei R (2020) Social movie recommender system based on deep autoencoder network using Twitter data. *Neural Comput Appl*, 1–17
21. Azimdoost B, Westphal C, Sadjadpour HR (2013) On the throughput capacity of information-centric networks. In: *Proceedings of the 2013 25th international teletraffic congress (ITC)*. IEEE, pp 1–9.
22. Xiao C, Mei J, Müller M (2018) Memory-augmented Monte Carlo tree search. In: *AAAI*, pp 1455–1462
23. Karami A, Gangopadhyay A, Zhou B, Kharrazi H (2018) Fuzzy approach topic discovery in health and medical corpora. *Int J Fuzzy Syst* 20(4):1334–1345

Enhancing Protection Against Scalper Bots with ML



R. Gowthamani , K. Sasi Kala Rani , C. R. Jayanth, B. Jeba Regan Raj, and J. Binesh

Abstract In this modern fast-moving digital world, Online Shopping has become an important part. When these online shopping sites announce an offer for a limited amount of time to a product that has high demand, customers will rush to buy the products for diluted prices. But unfortunately, there are some other people called scalpers, who will buy those products in large quantities and sell them in the market for higher prices to make a profit. In order to buy those products faster, they are using some bots called Scalper bots. Using this, they can buy a large number of products in a short interval of time which is much faster than normal humans. Thereby making legitimate customers lose their opportunity to buy the product. The main problem is that the scalping bots are very fast and intelligent enough to buy many products in a short amount of time and can even bypass some varieties of CAPTCHAs. The efficiency of the bot lies in its speed. If we are able to find whether the user is a human or a bot, we can extend the time required to perform the checkout/buying process for the bot. By doing so, we can reduce its efficiency tremendously. Our solution involves a Machine Learning Algorithm to find out the average minimum amount of time taken by a human to complete the checkout process and use that as a standard to determine whether a user is a bot or a human. The proposed system improves the accuracy by predicting bots with the help of advanced machine learning algorithms.

R. Gowthamani (✉) · K. Sasi Kala Rani · C. R. Jayanth · B. Jeba Regan Raj · J. Binesh
Sri Krishna College of Engineering and Technology, Coimbatore, India
e-mail: gowthamanir@skcet.ac.in

K. Sasi Kala Rani
e-mail: sasikalaranik@skcet.ac.in

C. R. Jayanth
e-mail: 19eucs049@skcet.ac.in

B. Jeba Regan Raj
e-mail: 19eucs053@skcet.ac.in

J. Binesh
e-mail: 19eucs024@skcet.ac.in

Keywords Big data · Classification · Binary classification algorithm · Machine learning

1 Introduction

In this modern fast-moving digital world, Online shopping is an important part of this modern world. Many people have started shopping online nowadays. Online shopping sites are also encouraging customers by announcing attractive offers now and then for a short amount of time to boost their sales. At that time, people will rush to the internet to buy their products. But there are some other people called Scalpers who will misuse this to gain money. They will buy a large number of products that have high demand in the market and later when the product has become out of stock, they will sell this for a high cost. Thus, it makes legitimate customers lose their opportunity to buy the products and they are forced to pay the price determined by the scalpers. Not only e-commerce sites, but scalpers are also attacking many online marketing sites which affect the film industry, music industry, and many other industries [1]. This not only affects the customer, but also the seller. To buy these products, Scalpers are using Programmed Bots called Scalper Bots which are very fast in checkout processes. These bots are fast and intelligent enough. Inserting CAPTCHAs seems to be a solution to stop these scalpers. But usually, CAPTCHAs are inserted at the last step of all the processes, i.e., usually at the payment page. So, the scalpers take this as an advantage and use the bot to fill all other fields like selecting the product, logging into the account, filling the address fields, etc., except the last page so that the scalper can just solve the CAPTCHA and buy the product. But normal users on the other hand have to fill all the fields manually and they also have to solve the CAPTCHA as well which will make the condition even worse. This could be a major issue for legitimate customers [2]. As there aren't enough factors to distinguish between a human and a bot, it is almost impossible to detect bots. The only thing which differs between humans and bots is the time taken to do the checkout process. It is also the only advantage that a bot has over humans. Usually, bots are way faster than humans as they are automated to autofill all the data required for the checkout process. Using Machine Learning Algorithm, the minimum amount of time taken by a human to buy a product has to be found (standard time) by providing some data sets of some verified users [3]. By comparing that time with the time taken by each user to do the checkout process, we can determine if a user is a human or a bot. If any user is found to check out faster than the standard time, then they will be asked to solve CAPTCHAs just to increase the checkout time. Since the efficiency of the scalper bot lies in its speed, our proposed solution directly undermines the efficiency of these bots. Thus, giving humans a chance to buy the products.

2 Existing System

In this solution, they have proposed a method to prevent scalper bots from buying tickets for concerts. This solution is based on the fact that concert tickets are mostly bought by the people from the region where the concert takes place. If a large number of tickets are being bought by users from some other region, there is a high possibility that these users could be bots. By collecting geographic data when selling the ticket, we determine if the user is a bot or a human. Thus, we can stop them from buying tickets.

2.1 Disadvantages

This technique can be used only for events like concerts and matches in which the geographic location is known. But in the case of online shopping, we cannot use the location as a factor because users can buy a product from almost any part of the world [4]. This solution doesn't work well with scalpers near the event location. Most of the time, scalper bots use VPN to hide their IP address. Bots can still buy tickets using VPN near the event location [5, 6].

3 Proposed System

3.1 Block Diagram

Time taken by normal users to perform the checkout process is used to train the machine learning model. The classification algorithms play a vital role while classifying humans from bots. Machine learning algorithms can be used to find bot in an efficient way. The time spent by the user inside the website to buy the product is noted, before redirecting the user to the payment portal, the time taken by him/her will be sent as input into the trained model. Based on its prediction, the user will be either redirected to the payment portal instantly (In case of humans) or made to wait for a few seconds and then redirected to the payment portal (In case of bots). Figure 1 explains the architecture of the overall system. This waiting time will slow down the bot so that scalpers can no longer gain an advantage by using such bots. This part is versatile, i.e., instead of making the bot wait for a few seconds, we can either give a puzzle to it to solve.

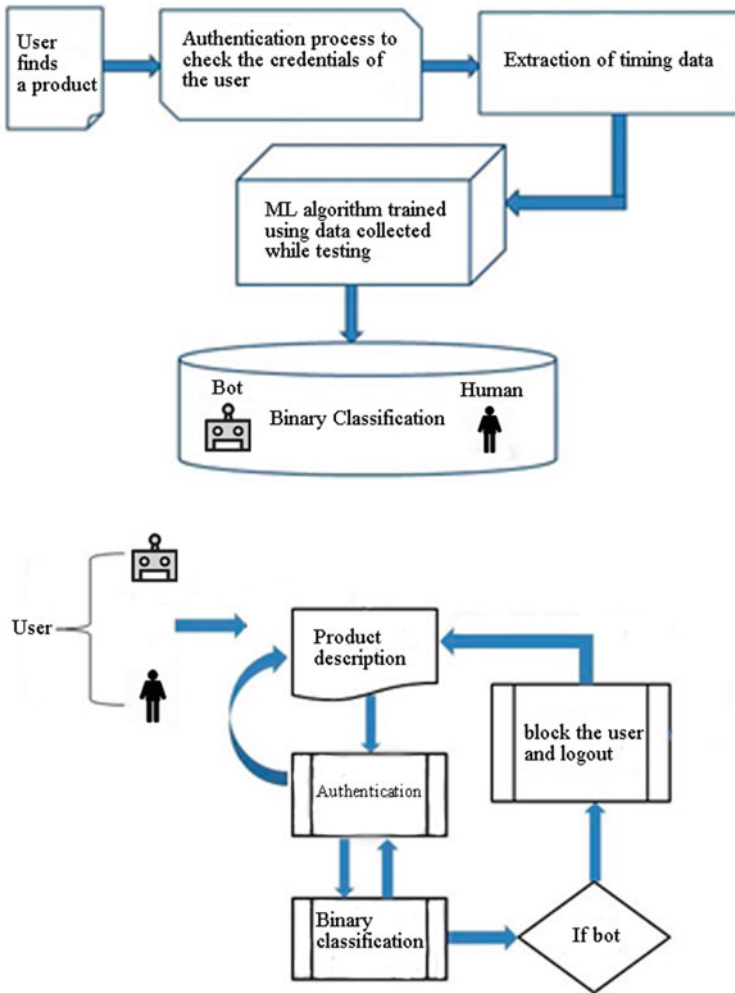


Fig. 1 Block diagram

3.2 Collecting User Timing Data

This stage requires the collection of timing data of users. The time taken by humans to buy a product can be easily obtained by just asking some people to do it. A checkout bot is designed by us and made to do the checkout process in the e-commerce website which we made to collect the timing data.

3.3 Training the Machine Learning Algorithm Using the Collected Data

The collected data is cleaned to remove any repeated data set and to remove any empty columns and then each timing data is marked whether it is a bot or human's timing. Then it is fed into the Machine Learning Model so that the model can be used to make predictions.

3.4 Validating User

Now the trained machine learning model can be used to validate if a user is a human. We have to pass the checkout timing for that particular user to the Machine learning model and it classifies if it is a human or not. Once a user is found to be a bot, we can increase the checkout process timing by making the user wait for some time before the payment process. Thus, it removes the advantage of bots over humans.

3.5 Improving the Accuracy of the Model

The accuracy of the model can be improved by including the time period spent by users on each page in the data set which will be used to train the Machine learning model so that it can make predictions more precisely.

4 Implementation

4.1 Web Application

An e-commerce web application made using the Django framework is designed to demonstrate the working of our solution. It is also used to get the dataset to train our model.

The web application which we have designed contains some common pages found on famous e-commerce sites like amazon, flip kart, etc. All these web applications mostly have a homepage that has a list of products. After selecting our products, we will be forwarded to the login page where we have to login or register. After logging into our account, we will give our basic information like Name, Address, and Mobile number in order to buy the product. On the next page, we will select our mode of payment and fill our card details. Now we are one step away from buying our product. After confirming, the product will be purchased. The Web App which we have designed also has most of these features in order to get a dataset to train our

model [7]. The user's timing will be started to note down as soon as the home page loads. After going through all the processes one by one, the user's timing up to the "Payment Method" will be calculated and sent to the model. Based on its prediction, the user is either allowed to fill the payment details instantly or the user will be made to wait for certain time and then allowed to fill the details [5, 8].

4.2 Checkout Bot

A bot is made using a python program to checkout products automatically. It will be used to collect the timing data set of bots using the e-commerce web application we have designed [6]. Based on the checkout bots designed by the scalpers, we have designed a checkout bot using python programming language, which will do all the processes from selecting the product to buying that product. Depending on the sequence of pages and input in the Web App, we have to configure the bot before executing it. We also provide the info which has to be filled in the input fields which includes the passwords, card details, and so on. In this, we have used fake information and not our original details. We have to start the Web Application and we have to start the bot as well. When the bot completes one iteration (from the home page to the payment page), its time is noted down, and then using the `openpyxl` module, it is stored in a sheet. The bot's time may vary based on the network traffic. A number of data sets are collected in this way [9]. After this, time taken by the normal user is also calculated in the same way but instead of the bot, we did those checkout processes. Finally, our dataset contains the timestamps and another column contains the answer if the user is a bot or human which is represented using 1 for bot and 0 for human.

4.3 Bot Detection Model

A binary Classification Algorithm is used in this project to differentiate between human and bot. It comes under classification algorithm which is a Supervised Learning Technique which makes predictions based on the data given to it [10]. The output will be in the form of Binary, i.e., either yes or no, 1 or 0. In this case, it predicts if the user is a Bot or not. The steps involved in this model are:

1. The CSV file containing the datasets is imported and cleaned to remove duplicate values.
2. The data set is splitted into two in the ratio 8:2 in order to train and then test the model
3. Then the model is trained. After that, it is tested to check if it is working properly. Some adjustments are made based on the prediction accuracy

4. The Python file containing the model was not created/embedded inside the App. It was created outside, trained, and tested and that trained model is stored as a module
5. That trained module is imported into the App so that we don't need to train the model each time we start the server
6. Now when a user tries to buy a product, those timestamps will be collected and sent to the model, based on its prediction, and the consecutive steps are followed.

5 Result

5.1 *E-commerce Web Application*

Once the user enters the e-commerce website, a timer will be started to note down the time. The user will be selecting the products which he/she wants and adds them to the cart then logs into the account. After that, the user will proceed to the payment process in which the user has to enter his address and some more personal information. In the next step, the user has to perform the payment to buy the product, i.e., right now the user is one step away from buying the product. Now the timer will be checked to find how much time the user has spent doing all these processes. This time is sent to the model to make a prediction if the user is a bot or a human.

5.2 *Detected a Human*

We use a binary classification algorithm which is trained using datasets collected from bots and verified humans to detect if the timing data obtained from the timer is that of a bot or a human. The above image is an example of the result of the binary classification algorithm. The timing is too high to be considered as a bot. So, the classification algorithm classified it as a human.

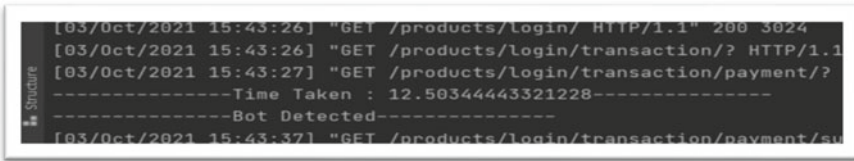
5.3 *Detected a Bot*

In the above image, the classification algorithm classified the user as a bot as the timing is too low to be considered as a human. Once a bot is detected, we can slow down its checkout process to such an extent that its checkout timing comes close to that of a human. Figures 2 and 3 represents the detection of human and bot identified by the proposed model. As soon as the user opens the product page, the timer starts and it will record the time taken by the user to do the checkout process. At the end of their checkout process before payment, the time taken so far will be used to determine if it is a user or not. If the user is found to be a bot, it will be made to wait for some



```
[03/Oct/2021 15:36:52] "GET /products/login/transaction/? HTTP/
[03/Oct/2021 15:36:55] "GET /products/login/transaction/payment
-----Time Taken : 25.498348474502563-----
-----Human-----
[03/Oct/2021 15:37:11] "GET /products/login/transaction/payment
```

Fig. 2 Output represents human



```
[03/Oct/2021 15:43:26] "GET /products/login/ HTTP/1.1" 200 3024
[03/Oct/2021 15:43:26] "GET /products/login/transaction/? HTTP/1.1
[03/Oct/2021 15:43:27] "GET /products/login/transaction/payment/?
-----Time Taken : 12.50344443321228-----
-----Bot Detected-----
[03/Oct/2021 15:43:37] "GET /products/login/transaction/payment/su
```

Fig. 3 Output represents bot

time before the payment process. By doing so, we undermine the efficiency of these bots. Thus, allowing normal human users a great opportunity to buy the product.

5.4 Comparison Chart

The graph shows the quantity of products bought by the bot in the absence of a bot detection mechanism and in the presence of our bot detection mechanism for a given amount of time.

From the graph, we can see that the efficiency of the bot is tremendously reduced by making it to wait for a certain amount of time. Figure 4 shows the accuracy of our proposed work.

6 Conclusion

The proposed system can be used to eliminate the threat of scalper bots. The Graph shows that the quantity of products bought by the bots is reduced very much. It tremendously reduces the efficiency of these bots. By delaying these bots to buy the products, a lot of normal customers get the chance to buy those products. Our Proposed Model uses the binary classification algorithm to find bots with the help of a timer. If the user is found to be a scalper, then the user is made to wait for a few seconds. In our solution, we have made the bot wait for a few seconds because we found it is simple and reliable. But it is the choice of the developer, i.e., instead of making them wait for a few seconds alone, we can send puzzles to solve or some

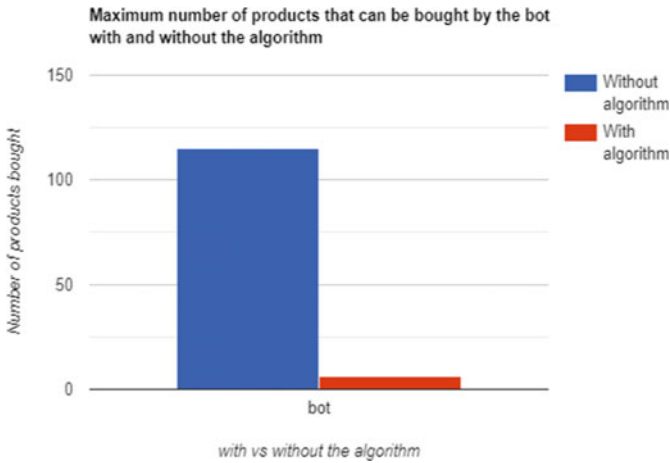


Fig. 4 Graph showing the number of products bought by the bot with and without our bot detection mechanism

other mechanism to the users who are suspected to be a bot instead of making them wait for few seconds. Thereby we can not only delay the bots from buying, but we can also prevent the product from falling into the hands of the scalpers which will be used by them later to gain money.

References

1. Kaiser E, Feng W-C (2010) Helping ticket master. In: Computer communication workshop, May 2010
2. Iliou C, Kostelac T, Tsirikla T, Katos V, Vrochidis S, Kompatsiaris I (2021) Web bot detection international conference in cyber security, September 2021
3. Lin C-C, Hsiao H-C. Bot-enabled ticket scalping. Criminal Investigation Bureau, National Policy Agency, Taiwan, National Taiwan University, Taiwan
4. Nassar N, Miller G (2012) Method for a two dimensional honeypot to deter web bots in commerce systems. In: Fourth international conference on computational aspects of social networks (CASoN)
5. Wasnock N (2018) Ticketmaster verified fan, backstage pass
6. Brock S (2021) Scalping in ecommerce: ethics and impacts, February 2021
7. Joe DeMars University of the Pacific, The Secondary Market: A Model of Misdirection, Backstage Pass (2018)
8. Corey S-L (2014) The truth about ticket scalping laws
9. Aura T, Nikander P, Leiwo J (2000) DoS-resistant authentication with client puzzles. In: Workshop on Security Protocols, April 2000
10. Gowthamani R, Sasi Kala Rani K, Indira Priyadarshini M, Rohini M, Ebenezer G, Thomas E (2022) Web based application for healthy habit development through gamification with ML. In: 2022 4th international conference on smart systems and inventive technology (ICSSIT), pp 1338–1345. <https://doi.org/10.1109/ICSSIT53264.2022.9716318>

An Enhanced Optimized Abstractive Text Summarization Traditional Approach Employing Multi-layered Attentional Stacked LSTM with the Attention RNN



M. Nafees Muneera  and P. Sriramya 

Abstract Progress in technology is occurring at an escalating speed all through this world. The human concentration stretch remains continually lessening, and the time span an individual needs to put in reading is decreasing at a rapid pace. Due to the development of technology, a wide amount of text data are produced daily through social networking services, business firms, medical information services, and news. In fact, it remains a complicated job to excerpt fascinating designs out of the text data like views, summaries, and actuality particulars possessing variable length. Due to the issues of the length of text data and the complication of feature value excerption in news, this study proffers a hybrid feature excerption out of the multi-layered attentional StackedLSTM (long short-term memory) (MASta-LSTM-RNN) alongside the Attention RNN network which automatically produces a summary out of a lengthy text. In this study, an input of the StackedLSTM module and the correlation betwixt words and classes are portrayed by a discrete vector as an input of the MLP module for the sake of attaining exhaustive learning of spatial attribute data, time-series attribute data, and correlation betwixt words and classes of news text. The correlation between words is portrayed by the word vector. For inspecting the steadiness and execution of the proffered methodology, several experiments were executed. The outcomes of these experiments exhibit that the proffered methodology resolves the issues of text length, the complication of attribute excerption in the news text, and categorization of news text effectually and has acquired finer results.

Keywords Textsummarization · Abstractive · Long short term memory · Stacked LSTM · ROUGE-RNN

M. Nafees Muneera (✉) · P. Sriramya
Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India
e-mail: nafeesmuneera@gmail.com

1 Introduction

Presently, there remains a wide amount of textual information present inclusive of online archives, articles, news, and reviews, which consist of lengthy strings of text that required being summarized [1]. The significance of text summarization is because of various causes such as the recovery of notable data out of a lengthy text in a brief duration, simple and swift loading of much significant data, and solving of the issues related to the parameters required for summary assessment [2]. Because of the progression and development of automated text summarization methodologies that have given notable outcomes in several languages, these methodologies are required to be analyzed and summarized. Hence, in the present study, we surveyed the latest methodologies and concentrated on the approaches, datasets, assessment standards, and adversities of every technique alongside the way whereupon every methodology discussed difficulties.

Text summarizing is used in applications like search engines and news web pages [1]. Previews are generated as snippets in search engines, and news web pages produce headlines for defining news to make wisdom retrieval easier [3, 4]. According to genre, kind of summarizer, number of documents, and function text summarizing can be divided into several classes [5]; one text summarization classification technique divides the summarization operation into abstractive classes [6].

Since Abstractive text summarization (ATS) needs comprehension of the document for producing the summary, wide natural language processing (NLP) and state-of-the-art machine learning approaches are needed. Permissible summarization must possess the enduring: sentences that manage the arrangement of the chief conceptions and notions given in the initial text, minimum to not repetitive, sentences that are even and rational, and the capacity to recall the sense of the text, for lengthy sentences too [7]. Moreover, the produced summary should be concise when expressing significant data regarding the initial text [2, 9].

Deep learning approaches are used in abstractive text summarization foremost ever in 2015 [8] and proffered paradigm relied upon encoder-decoder structure. To such applications, deep learning approaches gave outstanding outcomes and are widely used recently.

Abstractive Text Summarisation (ATS) schemes are crafted by implementing any text summarization techniques: abstractive or hybrid. The preprocessing technique chooses the much significant words out of input text and employs them for producing a summary. The abstractive technique portrays input text in an intermediary format that produces the summary having words and sentences, which contradict the initial text sentences. The hybrid technique joins abstractive techniques. The enhanced categorizations for the ATS system employs in categorizing the text information.

The common structure of ATS system, illustrated in Fig. 1, includes ensuing assignments: (i) Pre-processing: generating an organized portrayal for initial text [9] employing several linguistic approaches such as Noise Removal, elimination of stop-words, words tokenization, stemming and part-of-speech tagging, and so on, (ii) Processing: employing one among the text summarization techniques

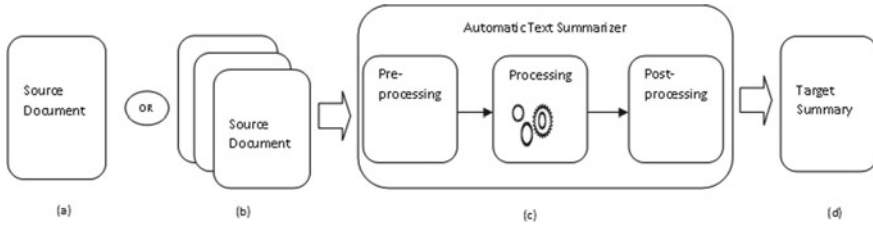


Fig. 1 **a** Source-document, **b** Multi-document, **c** Automatic text summarizer, and **d** Target summary

by implementing an approach or many to transfer the input document(s) to the summary, and (iii) Post-Processing: resolving a few issues across produced summary sentences such as anaphora resolution and rearranging the chosen sentences formerly producing the last summary.

2 Associated Studies

Raphal et al., analyzed many ATS procedures in common [10]. Their work distinguished betwixt various paradigm structures like reinforcement learning (RL), supervised learning, and attention operation. Moreover, correlations concerning word embedding, data processing, preparation, and confirmation were executed. Nevertheless, there remain nil correlations of the quality of many paradigms which produced summaries.

Additionally, the two extractive and abstractive summarization paradigms were summarized in [11, 12]. In [11], the categorization of summarization works relied upon three factors: input, purpose, and output. And this research analyzed exclusively five abstractive summarization paradigms each one, and this research analyzed exclusively five abstractive summarisation paradigms each one. Contrastingly, concentrated upon the datasets and preparing approaches besides the structure of many abstractive summarisation paradigms [12]. Nevertheless, the attribute of the produced summary of various approaches and the assessment standards have not been articulated.

Shi et al., gave an exhaustive survey of many ATS paradigms that are relied upon seq2seq encoder-decoder structure for convolutional and RNN sequence-to-sequence paradigms. The concentration was on the architecture of the network, preparing scheme, and programs used for producing the summary [13]. Even though many works examined abstractive summarization paradigms, some works have executed exhaustive research [14]. Additionally, many of the former surveys included the approaches up till 2018, although surveys have been publicized in 2019 and 2020 [11, 12]. Herein the study, we discuss many latest deep learning-based RNN ATS paradigms. Moreover, the study remains the foremost one to discuss the latest approaches implemented in abstractive summarization like Transformer.

(a) This work ensues the experimentation notion of the amalgamation of DL approaches-based news text categorization and chooses CNN, LSTM, and MLP paradigms for proffering a custom MLP paradigm of news text categorization relied upon dual input joined depth learning. (b) The proffered paradigm is conveyed in parallel having word vector and word dispersal. The correlation between words is portrayed by a word vector as the input of the CNN module, and the correlation between words and classes is portrayed by a discrete vector and an MLP module input to recognize exhaustive learning of spatial attribute information, time-series attribute information, and correlation among words and classes of news text. (c) Several experiments were carried out for inspecting the steadiness and execution of the proffered methodology. (d) The experiment outcomes reveal that the execution of the proffered methodology is much finer when compared to the rest of the techniques concerning anticipated precision and the rest of the execution standards.

This work gives an outline of the techniques, datasets, assessment standards, and adversities of deep learning-based ATS, and every title has been addressed and examined. The study categorized techniques according to output kind into solitary-sentence summary and multiple-sentence summary techniques. Additionally, inside every categorization, we correlated betwixt the techniques relating to structure, dataset, dataset preparation, assessment, and outcomes.

The rest of the study is ordered as follows. Section 3 defines the proffered method in depth. Section 4 describes various preprocessed ATS approaches and Sect. 5 examines various approaches alongside building blocks for applying ATS system along it with discusses the latest solitary-sentence summarization techniques, whereas the multiple-sentence summarization techniques are also included. Section 6 examines datasets and assessment standards accordingly with discussion. Conclusion is given in Sect. 7.

3 Proffered Method

The ensuing block drawing in Fig. 2 portrays the performing architecture of the proffered method. At first, the input dataset out of CNN/Daily Mail is prepared out of the input database in which the preparing process has to endure the preprocessing, attribute excerption out of the preprocessed information, and preparing the abstracted information for categorizing the summarized abstractive information. In preprocessing, the input data is preprocessed employing Noise Removal, Word Tokenization, Stemming, Removal of Stop Words, and POS (Parts of Speech); later, the input data is prepared for attribute excerption. The prepared text data should endure the attribute excerption by proffered structure paradigm MASTa-LSTM-RNN information for wide-ranging preparation of the multiple sentences, and later. The execution of the summarized data is analyzed by the ROUGE score and by the execution standards.

In this section, the proffered MASTa-LSTM-RNN paradigm is explained in depth. MASTa-LSTM-RNN utilizes the strength of abstractive approaches to generate a

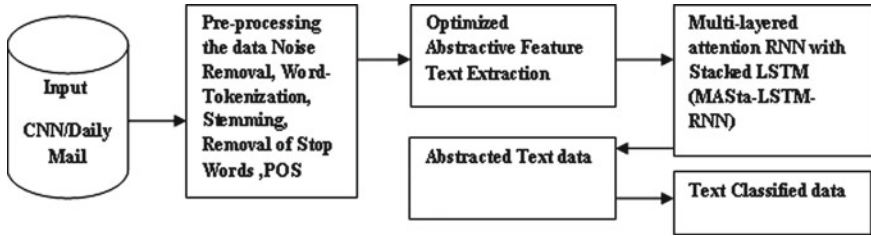


Fig. 2 The proffered structure of the ATS employing deep learning conception

brief summary. The hybrid approach is later employed for generating a more appropriate headline that matches the produced summary. We start by excerpting important sentences out of a comprehensive article employing our proffered procedure mentioned hereunder ensued by the abstractive summarization of the chief sentences employing a reinforced abstractive mechanism.

4 Preprocessing

4.1 Preprocessing Approaches

There remain several preprocessing approaches such as noise elimination, sentence dissection, word tokenization, and so on. Several of these approaches are generally employed in preprocessing stage of the ATS system.

4.2 Noise Elimination

It eliminates inessential text from the input document such as the header, footer, and so on [15].

4.3 Sentence Dissection

It breaks up the text into sentences [16]. In a number of situations, using end markers like “.”, “?”, or “!” to divide sentences is improper. When we rely on the markers, terms like “e.g.”, “i.e.”, “4.5”, “Mr.”, “Dr.”, or “etc.” lead to incorrect sentence border detection. Easy heuristics and common formulations have been used to solve the problem [15].

4.4 Elimination of Punctuation Marks

Punctuation marks have long been thought of as distracting terminology seen within the text. As a result, removing them is very supportive of previously performing various NLP works [16].

4.5 Word Tokenization

It deconstructs text into component words.

White space, commas, dashes, dots, and other symbols will be used to separate words [15].

4.6 Named Entity Recognition (NER)

It detects words in the supplied text as names of things (that is position name, individual name, business name, etc.).

4.7 Elimination of Stop Words

These words appear often inside the text that includes pronouns, auxiliary verbs, articles, prepositions, and determiners. These have been eliminated since these by no means include whatever beneficial sense to examination [17] and possess nil impact on choosing significant sentences [16].

4.8 Stemming

It lessens words having a similar root or stem toward a general format by eliminating changeable suffixes [16] such as “es” and “ed”. The objective of stemming remains in acquiring stem/radix of every word for giving attention to its semantics [15].

4.8.1 Part-of-Speech (POS) Tagging

This assigns POS tags to words in a sentence (for example, verb, noun, and so on).

5 Abstractive Text Summarization Paradigms

This research aims to establish an ATS paradigm employing variant MASTa-LSTM-RNN-RNN. As the general format of LSTM, known as classical LSTM, possesses few restrictions such as reliance amidst cells that are not powerful [18, 19], instead of using the standard LSTM, the attention RNN with StackedLSTM-MASta-LSTM-RNN was used to create+ paradigm. Classical LSTM and MASTa-LSTM-RNN with the Attention RNN are thoroughly covered in this section.

5.1 Classical LSTM Unit

Long short-term memory (LSTM) remains a module of the recurring neural network, which could detect and recollect information designed to a specified time frame. LSTM obtains a series of text as input and anticipates a series of text as output. This contains memory cell, input gate, output gate, and forget gate. The above gates manage memory content and cell states in the present time code [20]. Structure of the classical LSTM unit is illustrated in Fig. 3.

The upper line is referred to as a memory pipe because it receives previous memory content c_{t-1} as input and outputs final memory c_t . Former memory content is transferred using two different procedures: bitwise multiplication (\times) with the output of the forget's gate and bitwise summation ($+$) with the input gate output. Forget gate produces output in extent zero to one that performs a significant part in determining the memory content in the present timecode. The output value nearer to 0 refers that much of the former memory content would almost be unremembered and the contradictory would occur for output value close to 1. The bitwise summation ($+$) of the input gate's output determines how much input from the current timecode can

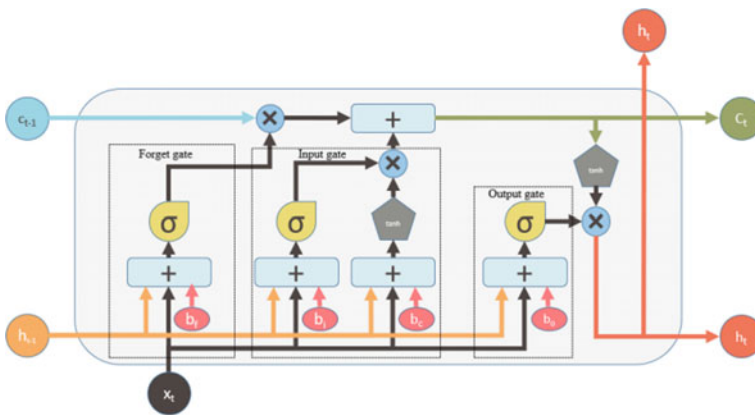


Fig. 3 Classical LSTM has 3 gates: input, forget, and output

be incorporated into the previous memory content for creating the final memory at the current timecode. In this, the input gate too generates output amidst an extent of 0–1.

The content of the memory block is administered by 3 gates: input, forget, and output gates. In this, c_{t-1} and c_t correspond to the contents of former and present memory cells, h_{t-1} and h_t correspond to the outputs of former and present states, x_t corresponds to the input vector, x corresponds to bitwise multiplication, $+$ corresponds to bitwise summation, \tanh corresponds to a hyperbolic tangent function, sigmoid function, and bf, bi, bc, and bo correspond to the bias of various.

Forget gate obtains output of former state h_{t-1} , input vector x_t , and bias b_f as input and produces value amidst an extent of 0–1, which determines what amount of content of memory would be transferred via memory pipe.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (1)$$

The input gate too produces a value between 0 and 1, which describes what amount of input out of the present timecode would include the memory content in the present timecode

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i), \quad (2)$$

$$C_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

Output gate obtains content h_{t-1} of former state, input vector x_t , and bias b_o as input and generates as output. Lastly, the content of the present state h_t is generated by employing the value of

$$O_t = \sigma(W_{x0}x_t + W_{h0}h_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \tanh(ct) \quad (5)$$

In the structure of the classical LSTM, therein remains nil associate betwixt a gate and its former memory cell. Due to this, it remains impossible to procure the former memory cell state while the output gate stays locked amidst preparation. A variant of LSTM, known as MASTa-LSTM-RNN, could subdue this issue.

5.2 Abstractive Summarization

A summarizer remains abstractive when it attempts to produce the ideal series $Y = (y_1, y_2, \dots, y_n)$, in which $y_i \in V$ Where y_1, y_2, \dots, y_n are the output sequence for the input sequence of the value in the encoder input sequence.

LSTM is a recurrent neural network module having vast implementations in sequence prognosis issues. While the output gate of a general LSTM format, known as classical LSTM, is locked, it is unable to retrieve the content of its former memory cell. By including additional connection betwixt every gate and former memory content, the issue of classical LSTM could be resolved. That extra link, called peephole connection, permits entire gates for employing the former memory cell content although while output gate is locked. The structure of MASTa-LSTM-RNN is illustrated in Fig. 4.

Every gate is connected to the previous memory cell c_{t-1} in this scheme. Aside from h_{t-1} , x_t , and bias, every gate has been provided the memory of the previous cell as an input. That permits procuring content of former memory cell while the output gate is locked.

$$i_t = \sigma (W_{xi}x_t + W_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i) \tag{6}$$

$$f_t = \sigma (W_{xf}x_t + W_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f) \tag{7}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{8}$$

$$o_t = \sigma (W_{xo}x_t + W_{ho}h_{t-1} + w_{co}c_t + b_o) \tag{9}$$

$$h_t = o_t \tanh(c_t) \tag{10}$$

In this, the former memory cell content c_{t-1} besides the rest of the criteria is given as input in peephole convolutional LSTM when classical LSTM in no way regards c_{t-1} as input. Considering c_{t-1} as input in the multi-layered attention with StackedLSTM (long short-term memory) (MASta-LSTM-RNN) affects outcome of sequence.

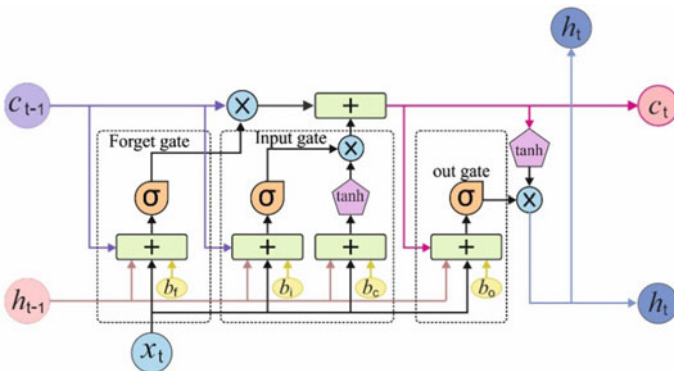


Fig. 4 MASTa-LSTM-RNN having a multi-layered connection

5.3 MASTa-LSTM-RNN Paradigm

This work aims to create an abstractive summarization paradigm employing MASTa-LSTM-RNN. Figure 5 displays the execution procedure of the MASTa-LSTM-RNN paradigm. We include multiple layers of LSTM in MASTa-LSTM-RNN that refer to that this possesses greater than a single hidden layer. We employ dual hidden layers for creating a paradigm because greater than three drops occur in the execution of the paradigm owing toward gradient decay through layers. At first, provided input text is transferred via the embedding layer, which transforms provided text as arithmetic dispensed portrayals. Such portrayals will be transferred via dual hidden layers of MASTa-LSTM-RNN, every layer having century hidden units. The encoder and decoder of the seq2seq paradigm are implemented in this. The encoder captures the attribute vector that represents the supplied text. Simultaneously, it calculates the attention weight for each provided word and saves it in an attention vector, which is then used to provide an accurate summary in the dense layer. In this case, each word's attention weight is computed and saved in a comparable attention vector. The attention scheme assists sequence-to-sequence paradigm in recalling significant qualities of a provided input [21]. While outputting a word, the attention weight of the input word at location t can be calculated as

$$ay_{t'}(t) = \frac{\exp(h_{xt}Th_{yt'})}{\sum \exp(h_{xt}Th_{yt'})} \quad (11)$$

in which $\exp(h_{xt}^T)$ depicts the final hidden layer produced subsequent to processing t th input word, and $h_{y_{t'}}$ depicts the final hidden layer produced out of the present stage of decoding. Attention operation computes the conditional possibility of an input word. When all of the provided text is transferred via the encoder, the decoder obtains the attribute vector out of the encoder and produces the finest text for a planned summary. Text produced by the decoder is transferred via the last dense layer, which employs the attention vector for choosing the much significant portions as the last summary. We implement our improved MASTa-LSTM-RNN paradigm on CNN-Daily Mail data set.

6 Results and Discussion

The CNN/Daily Mail dataset is a subset of the DeepMind Q&A dataset, which was created in 2015 and contains around 287 K news stories with 2–4 summary sentences per story. The collection is made up of multi-sentence summaries and online news pieces (781 words on average) (3.75 sentences or 56 tokens on a mean). There are almost 287,226 training couples, 13,368 validation couples, and 11,490 testing couples in this ready variant of the dataset. The Precision, Recall, and F-Measure

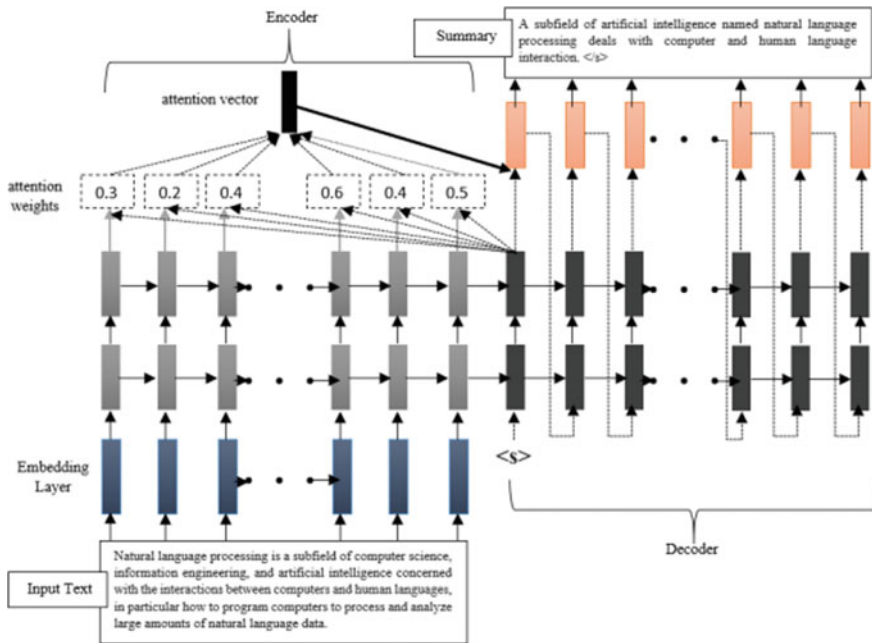


Fig. 5 The comprehensive plan of the proffered architecture that begins by obtaining input text and ends by producing an abstractive summary

scores, (i) Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [22], and (ii) Basic Element (BE) [23] are examples of commonly used assessment standards in the literature.

Precision Score Metric: This can be calculated by splitting amount of sentences present in the two reference and candidate (that is, system) summaries by amount of sentences in the candidate summary as depicted in Eq. (2) [24].

$$\text{Precision } P = \frac{|S_{\text{ref}} \cap S_{\text{cand}}|}{|S_{\text{cand}}|}$$

Recall Score Metric: This can be calculated by dividing the amount of sentences present in the two reference and candidate summaries by the amount of sentences in the reference summary as depicted in Eq. (3) [24].

$$\text{Recall } R = \frac{|S_{\text{ref}} \cap S_{\text{cand}}|}{|S_{\text{ref}}|}$$

F-Measure Score Metric: This is a measurement, which joins recall and precision metrics as depicted in Eq. (5) [24]. F-measure remains harmonic mean betwixt precision and recall.

$$\text{F-Measure } F = \frac{2(\text{precision})(\text{Recall})}{\text{Precision} + \text{Recall}}$$

ROUGE Metric: This remains the widest generally employed tool for the automated assessment of automatically produced summaries. ROUGE remains an array of metrics and software packages employed for assessing automated summarization and machine translation software in NLP [15]. This correlates with computer-generated summaries opposing many man-generated reference summaries. The fundamental notion of ROUGE is to calculate the amount of overlying units betwixt candidate (or system) summaries and reference summaries like overlaid n-grams. ROUGE is confirmed to be effectual in computing attributes of summaries and associates splendidly with man’s decisions. Several ROUGE metrics are presented as follows.

Table 1 signifies the correlation of present SUMMARUNNER, Optimized MAPCoL, SHEG + CONV (Excerption) having proffered MASTa-LSTM-RNN.

The above Fig. 6. is the overall comparison with the ROUGE-1, ROUGE-2, and ROUGE-L with the execution metrics and determines projected methods outperform the highest performance measures values.

Figure 7 shows the analysis of ROUGE scores where the x-axis indicates 3 types of ROUGEs, and the y-axis shows the score in percentage. The analysis shows that MASTa-LSTM-RNN has 43.75% in ROUGE-1, 18.5% in ROUGE-2, and 37.4% in ROUGE-L. Table1 indicates the comparison of the ROUGE score of summary with the variable length for the proposed MASTa-LSTM-RNN dataset (CNN/Daily Mail dataset).

Headline Production

The last stage of MASTa-LSTM-RNN is generating a headline on the abstractive summary generated in the former stage. To achieve this work, we engaged in sequence prognosis for generating the planned headline.

Table 1 ROUGE score computation for the dataset (CNN/daily mail dataset)

Methodologies	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)
Bottom-up sum	41.22	18.68	38.34
Summa runner	39.60	16.20	35.30
Optimized MAPCoL	41.21	21.30	39.4
SHEG + conv (extraction)	42.5	17.6	35.6
MASta-LSTM-RNN	43.75	18.5	37.4

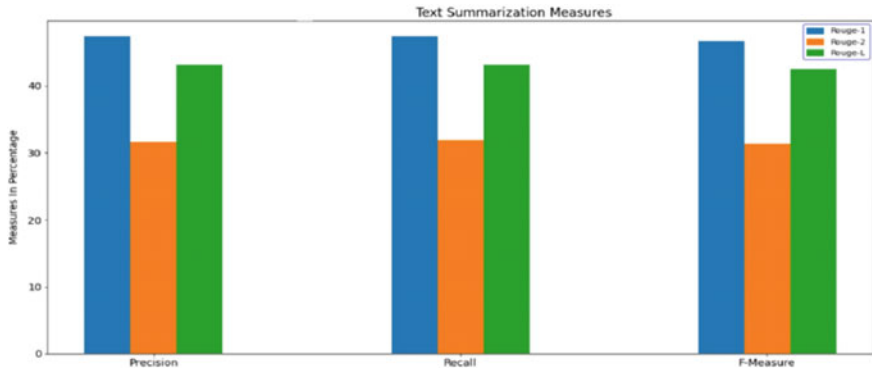


Fig. 6 Correlation of Execution metrics of Precision, Recall, and F-Measure alongside ROUGE-1, ROUGE-2, and ROUGE-L

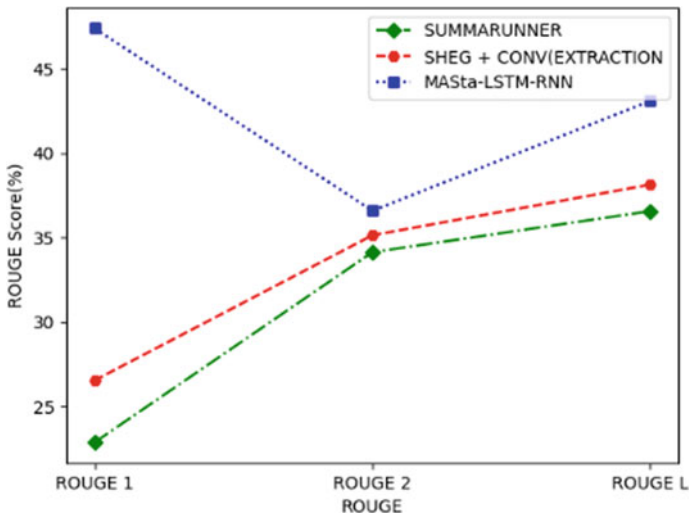


Fig. 7 Comparison of ROUGE scores

Extractive Summary

Tripoli libya rebels tripoli furiously hinting signs longtimelibyan leader moammar-gadhafi exploring network tunnels bunkers built beneath massive compound cnn sara sidner got peek passageways friday dubbed gadhafi inner sanctum correspondent covering battle tripoli walked steps pitch dark tunnel used flashlight navigate underworld described massive far said rebels cleared meters underground passage tunnel network believed extend way city international airport rixos hotel journalists two foreign nationals held five days pro gadhafi forces also thought extend neighbourhood gadhafi forces lobbing shells recently toward compound taken rebles tunnels sidner

saw wise enough adults walk side side spotted golf cart easily fit corridors sidner also saw range sights strolled labyrinth thick wall massive door sturdy lock charred ceiling couches beds fire apparently occurred pieces metal shrapnel section nato bomb fell roof caved another room contained videotapes lined shelf part tv studio gadhafi may recorded messages set like survival bunker sidner said air report literally city.

Reference Summary

Sostokcnn sara sidner sees another world in tunnel below tripoligadhafi may may have recorded his taped messages in studio there rebels are methodically searching through the winding passages eastok.

SummaRunner Summary

Tripoli libya rebels tripoli furiously hunting signs longtimelibyon leader moammargadhafi exploring network tunnels bunkers built beneath massive compound cnn sara sidner got peek passageways friday dubbed gadhafi inner sanctum correspondent covering battle tripoli walked steps pitch dark tunnel used flashlight navigate underworld described massive for said rebels cleared meters underground passages tunnel network believed extend way city international airport rixo. tripolilibya rebels tripli furiously hunting signs longtimelibyan leader moammargadhafi exploring network tunnels bunkers built beneath massive compound cnn sara sidner got peek passageways friday dubbed gadhafi inner sanctum corresponded covering battle tripoli walked steps pitch dark tunnel used flashlight navigate underworld described massive far said rebels cleared meters underground passages tunnel network believed extend way city international airport rixos s.

SHEG Summary

Tripoli libya rebels tripoli furiously hunting signs longtimelibyan leader moammargadhafi exploring network tunnels bunkers built beneath massive compound cnn sara sidner got peek passageways friday dubbed gdhafi inner sanctum correspondent covering battle tripoli walked steps pitch dark tunnel used flashlight navigate underworld described massive for said rebels cleared meters underground passages tunnel network believed extend way city international airport rixo. tripolilibya rebels tripoli furiously hunting signs longtimelibyan leader moammargadhafi exploring network tunnels bunkers built beneath massive compound cnn sara sidner.

MASta-LSTM-RNN

Sostokcnn sara sidner sees another workd in tunnel below Tripoli Gadhafi may have recorded his taped messages in studio there rebels are methodically searching through the winding passages eastok.

The above is the Input Dataset for the Text summarization of the given CNN-Daily dataset to get the abstractive and the extractive summary.

7 Conclusion

The created abstractive summarization paradigm employing multi-layered peep-hole convolutional LSTM attains finer execution when compared to other advanced paradigms concerning semantic and syntactic coherence. This created paradigm is enhanced by employing the central composite pattern alongside the response surface pattern. Anticipated ROUGE-1, ROUGE-2, and ROUGE-L scores identified through proffered methodology remain at 41.98, 21.67, and 39.84%, which are almost nearer to the experiment of 41.21, 21.30, and 39.42% accordingly. An improved MASTa-LSTM-RNN paradigm subdues a few issues, which are accompanying the prevailing ATS approaches. Semantic and syntactic coherence is also ensured in this established paradigm. Although this paradigm subdues a few issues of the rest of the paradigms, it also possesses few restrictions. This paradigm operates the least efficiently while we produce a lengthy text as a summary. This paradigm operates least efficiently to produce a lengthy text as a summary. We implemented this established and improved paradigm of the CNN-Daily Mail data set, and the MASTa-LSTM-RNN performs finer than the classical LSTM-based paradigms.

References

1. Allahyari M, Pouriyeh S, Assefi M et al (2017) Text summarization techniques: a brief survey. *Int J Adv Comput Sci Appl* 8(10)
2. Al-Saleh AB, Menai MEB (2016) Automatic Arabic text summarization: a survey. *Artif Intell Rev* 45(2):203–234
3. Turpin A, Tsegay Y, Hawking D, Williams HE (2007) Fast generation of result snippets in web search. In: *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information Retrieval-SIGIR'07*. Amsterdam, Netherlands, p 127
4. Trippe ED (2017) A vision for health informatics: introducing the SKED framework an extensible architecture for scientific knowledge extraction from data. <http://arxiv.org/abs/1706.07992>
5. Syed S (2017) Abstractive summarization of social media posts: a case study using deep learning. Master's thesis, Bauhaus University, Weimar, Germany
6. Suleiman D, Awajan AA (2019) Deep learning based extractive text summarization: approaches, datasets and evaluation measures. In: *Proceedings of the 2019 sixth international conference on social networks analysis, management and security (SNAMS)*. Granada, Spain, pp 204–210
7. Al-Radaideh QA, Bataineh DQ (2018) A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms. *Cogn Comput* 10(4):651–669
8. Rush AM, Chopra S, Weston J (2015) A neural attention model for abstractive sentence summarization. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, Lisbon, Portugal
9. Gupta V, Lehal GS (2010) A survey of text summarization extractive techniques. *J Emerg Technol Web Intell* 2(3):258–268
10. Raphal N, Duwarah H, Daniel P (2018) Survey on abstractive text summarization. In: *Proceedings of the 2018 international conference on communication and signal processing (ICCSP)*, pp. 513–517, Chennai
11. Dong Y (2018) A survey on neural network-based summarization methods. <http://arxiv.org/abs/1804.04589>

12. Mahajani A, Pandya V, Maria I, Sharma D (2019) A comprehensive survey on extractive and abstractive techniques for text summarization. In: Hu Y-C, Tiwari S, Mishra KK, Trivedi MC (eds) *Ambient communications and computer systems*, vol 904. Springer, Singapore, pp 339–351
13. Shi T, Keneshloo Y, Ramakrishnan N, Reddy CK (2020) Neural abstractive text summarization with sequence-to-sequence models: a survey. <http://arxiv.org/abs/1812.02303>
14. Joshi A, Fidalgo E, Alegre E, de León U (2020) Deep learning based text summarization: approaches, databases and evaluation measures. In: *Proceedings of the international conference of applications of intelligent systems*, Spain
15. Gupta, VK, Siddiqui TJ (2012) Multi-document summarization using sentence clustering. Paper presented at the 2012 4th international conference on intelligent human computer interaction (IHCI)
16. Gambhir M, Gupta V (2017) Recent automatic text summarization techniques: a survey. *Artif Intell Rev* 47(1):1–66. <https://doi.org/10.1007/s10462-016-9475-9>
17. Jaradat YA, Al-Taani AT (2016) Hybrid-based Arabic single-document text summarization approach using genatic algorithm. Paper presented at the 2016 7th international conference on information and communication systems (ICICS)
18. Shi X, Chen Z, Wang H, Yeung D, Wong W, Woo W (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: *Proceedings of the 28th international conference on neural information processing systems*. MIT Press, Montreal, QC, Canada, pp 802–810
19. Colmenares C, Litvak M, Mantrach A, Silvestri F (2015) HEADS: headline generation as sequence prediction using an abstract feature-rich space. In: *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics; Human Language Technologies; Association for Computational Linguistics*, Denver, CO, USA, pp 133–142.
20. Lipton Z (2015) A critical review of recurrent neural networks for sequence learning. [arXiv:1506.00019](https://arxiv.org/abs/1506.00019)
21. Lopyrev K (2018) Generating news headlines with recurrent neural networks. [arXiv:1512.01712](https://arxiv.org/abs/1512.01712)
22. Lin C-Y (2004) ROUGE: a package for automatic evaluation of summaries. Paper presented at the workshop on text summarization branches out, Barcelona, Spain
23. Hovy E, Lin C-Y, Zhou L, Fukumoto J (2006) Automated summarization evaluation with basic elements. Paper presented at the the 5th conference on language resources and evaluation
24. Moratanch N, Chitrakala S (2017) A survey on extractive text summarization. Paper presented at the 2017 international conference on computer, communication and signal processing (ICCCSP), Chennai

Implementation of Machine Learning Methods on Data to Analyze Emotional Health



Vijaya Chandra Jadala , Sai Kiran Pasupuleti , S. Hrushikesava Raju , Srihari Babu Gole , N. Ravinder , and B. Sreedhar 

Abstract The design and implementation of the intelligent model based on machine learning methodologies on datasets to analyze emotional and mental health, majorly concentrating to measure, cognize, simulate, and react to human emotions. The data captured by the apparatus is always available to the user and provides contact to interact with patients in case of Emergency. It stores data for future analysis and for pattern identification, Experts will be able to use Artificial Intelligent based machines to augment the human capacity to detect, diagnose, and treat mental health issues. Machines have a significantly higher computational capacity than humans, creating assessments from large pools of patient's data that will eventually make informed clinical decisions. This article will examine AI applications on emotional and mental health, It also highlights the implications of using emotional AI in identifying, diagnosing, and improving mental health. We focused on the conceptual and simulation model for result analysis where data stored in the cloud is retrieved as per the requirement and analyzed. Different machine learning algorithms are implemented on the dataset and results are identified through graphical analysis.

V. C. Jadala (✉) · S. K. Pasupuleti · S. Hrushikesava Raju · S. B. Gole · N. Ravinder · B. Sreedhar
Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,
Greenfields, Vaddeswaram, Guntur 522502, India
e-mail: drvijayachandra@kluniversity.in

S. K. Pasupuleti
e-mail: psaikiran@kluniversity.in

S. Hrushikesava Raju
e-mail: hkesavaraju@kluniversity.in

S. B. Gole
e-mail: gsriharibabu@kluniversity.in

N. Ravinder
e-mail: ravindernellutla@kluniversity.in

B. Sreedhar
e-mail: sreedharburada@kluniversity.in

Keywords Machine learning · Emotional health · Mental health · Artificial intelligence · Data mining · Datasets

1 Introduction

The science and technology improved systematically in the field of intelligence machines, its approaches have transformed every field in medicine, creating sustainable, faster, and effective healthcare facilities based on different wearable apparatus. Optical heartbeat monitoring is a methodology adopted to identify the physical and mental health of patient based on sensor technology [1]. The advanced technology established on Internet of things for mental health monitoring through wireless technologies such as mobile and cloud provides data of the client in cloud storage or wireless communication systems. It generally uses different sensors such as photoplethysmography [2]. Innovative incorporation of cloud with different technologies such as Block Chain and IoT is a multifaceted task as they are different tremendously distinct mechanisms [3]. Their execution and application are appraised to be a supplementary benefit for constructing them as a significant module of the future technologies [4]. It is an automated apparatus with registers and microcontrollers that are utilized to identify, evaluate, and communicate information based on body signals. An analysis of AI applications and the impact on mental health are analyzed and calculated based on the simulations for a better understanding of impacts.

The new applications allow doctors to monitor patients and track health patterns. Artificial intelligence constructed intelligent technologies are currently being used to facilitate early disease recognition and allow better thought of disease development, enhance, or reduce medicine that is handling dosages, and expose different treatments. Data science provides a professional approach related to data analytics approach towards the mental health related to the actions such as emotional, psychological, and social well-being [5]. The data analytics is the process of analyzing raw information based on different types of analytics such as descriptive, diagnostic, predictive, and prescriptive. To this end, the present study implements such a qualitative approach because the objective is not to explain the impact of AI emotion intelligence based on generalizations but rather a deeper understanding and interpretations of the applicability in mental health. The method adopted here is known as the interactive evolutionary computation, where this application helps in diagnosing the mental health [4].

The cloud computing provides the storage sharing option where the data can be stored and can be retrieved from wherever we want, away from the industries. The IoT simplifies and enables the storage of results online with the help of cloud storage. The different cloud services that support the IoT are Azure or AWS cloud services, Many cloud services are chargeable as per the usage that is elasticity and we use for experimental purposes the cluster of storage of Thinkspak is a cloud for data storage and retrieval [6]. The different cloud environments that support the Internet of things remain salesforce, SAP, Thinkspak, CISCO, Thingworx,

oracle, IBM Watson, Google Cloud Platform, Azure, and Amazon web/services. The different concepts that support Internet of things areas are Internet of Things, wireless networks, and cloud computing emerging technologies, Internet of thing devices, Internet of Things Networking and communication, IoT Cloud, Amazon Web Services Internet of Things that of developing and deploying an Internet of things, Industrial Internet of Things on google cloud platform, Industrial Internet of Things markets and security. The mobile applications also support the cloud computing, where the results of Internet of Things can be given. The Internet of things even supports the embedded motors and sensors [7].

An analysis of AI applications and mental health is necessary to better understand the impacts. Therefore, this article will perform an in-depth analysis of the impacts of AI on mental health by highlighting ethical issues that may arise when integrating AI into mental health care services as part of delivery. Activity trackers are such devices that are part of Internet of things associated with the software, electronic devices, sensors, cloud, and mobile technologies [8]. A medical health record will be viewed and verified by nurses, doctors, diagnostics, and pharmacists. It is the common medium to inform the status of mental health of the patient, As the medical record is generated electronically, it provides an easy way for data analysis. Artificial intelligence technologies evaluate the mental health of the object based on the Electronic Health Records (EHRs) using the parameters such as depression, psychiatric illness, suicide ideation, and attempts. Even data analysis plays a major role based on the electronic health data sets using machine learning methodologies [9]. Health care management is the major task in the mental health, as records play a major role to identify a person's mental health situation [10]. Internet of things and cloud technologies combinedly created a system, where input is taken through wearable objects and sensors, the received signals are converted to data through the control system [11]. The captured data explain the recognized emotions of the humans. The data will be transferred to the cloud and stored as data set for future references [12]. The different steps involved in implementing machine learning in mental health are retrieving data sets from the library and data loading, data cleaning, encoding data, and implementation of different model evaluation algorithms.

2 Related Work

Wearable devices are smart health trackers that collect biological, physiologic, and behavioral data through intelligent sensors. These devices can be used to monitor mood changes that affect mental health. Integrative computer technologies have allowed for the use of innovative applications and programs in mobile health (mHealth) devices. These devices use a combination of technologies (i.e., AI, NLP, computer sensing technologies, and machine learning) to make them intelligent or smart. Mobile and wearable technologies, arguably the most ubiquitous AI application in mental and behavioral healthcare, exist in mobile phones and other wearable devices to track mood and behavioral changes [13].

The different mobile applications are available to communicate with the cloud and display the result on the mobile app, the most used mobile app is blynk, which is used to create virtual components that will be interacting with microcontroller boards. The different connecting devices and sensors use the cloud environment to control remotely. It is a new platform that builds interfaces to monitor the results provided by the sensors and microcontroller units. It even provides different integrated development environment components such as buttons, gauges, sliders, and widgets for user interfacing. It is compatible with major microcontroller devices and freely available in playstore. E-health mobile applications are different methods used by psychiatrists to avoid or prevent the suicides. The Internet-based mobile applications help the doctors or medical staff to know the level of patient's psychology. There are many other mobile apps that are available in playstore which are easily interacting with the microcontroller. The apps are used to analyze the data that is produced by the sensors through microcontrollers, which create interfaces to the project with help of various widgets available as tools for the application. It selects the port 8080 for the hardware network connection and it uses the secured socket layer and transport layer security for data transmission and communication. The different cloud servers help to communicate with the mobile application [14].

Machine learning technologies use "statistical and computational methods to construct more robust systems with an ability to automatically learn from data" [8, p. X]. These systems use previously collected and labeled mental health data to associate input features from several data contradictions and predict mental states of newly undetected observations. These machine learning models require human guidance through data training to effectively point out or predict psychological markers related to mental health issues. In the last decade, social media has grown into a major platform. Individuals with Internet access can share, post, and talk about their lives [15]. Most of these activities, online conversations, and posts can reflect a person's mental state, including depression, anxiety, and/or stress. As a result, social media platforms are used to monitor public mental health, model mental well-being, and detect warning signs that signify mental health issues. Other relative mental health programs predict mental health disorders like anxiety, depression, and stress using public data from social media platforms like Facebook and Twitter.

3 Experimental Analysis

For the experimental analysis, we took the dataset of mental health, where the data mining is a technology that can retrieve, manage, and manipulate data in dataset based on the machine learning algorithm, this discipline combines mathematical knowledge and statistical analysis [16]. As part of experimental analysis, we combined the knowledge of data mining and machine learning. The ML methodology is the highlighting concept on the qualities, attributes, and properties of the dataset. Python is the language used in the machine learning process, where we used Scikit-learn, it supports different machine learning categories such as classification and regression

```

1 [1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
    
```

Fig. 1 Snippet to import libraries at Jupyter notebook

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Timestamp	Age	Gender	Country	state	self_empt	family_hist	treatment_work	inte_no_empt	remote_u_tech	com_benefits	care_optiv	wellness_seek	help_anonymit	leave_mental_h	phys				
27-08-2014 11:29	37	Female	United	Stu	IL	NA	No	Yes	Often	Jun-25	No	Yes	Yes	Not sure	No	Yes	Yes	Somewha	No
27-08-2014 11:29	44	M	United	Stu	IN	NA	No	No	Rarely	More thar	No	No	Don't knor	Don't knor	Don't knor	Don't knor	Don't knor	Maybe	No
27-08-2014 11:29	32	Male	Canada	NA	NA	NA	No	No	Rarely	Jun-25	No	Yes	No	No	No	Don't knor	Somewha	No	No
27-08-2014 11:29	31	Male	United	Kir	NA	NA	Yes	Yes	Often	26-100	No	Yes	No	Yes	No	No	No	Somewha	Yes
27-08-2014 11:30	31	Male	United	Stu	TX	NA	No	No	Never	100-500	Yes	Yes	Yes	No	Don't knor	Don't knor	Don't knor	Don't knor	No
27-08-2014 11:31	33	Male	United	Stu	TN	NA	Yes	No	Sometim	Jun-25	No	Yes	Yes	Not sure	No	Don't knor	Don't knor	Don't knor	No
27-08-2014 11:31	35	Female	United	Stu	MI	NA	Yes	Yes	Sometim	03-May	Yes	Yes	No	No	No	No	No	Somewha	Maybe
27-08-2014 11:32	39	M	Canada	NA	NA	NA	No	No	Never	03-May	Yes	Yes	No	Yes	No	No	Yes	Don't knor	No
27-08-2014 11:32	42	Female	United	Stu	IL	NA	Yes	Yes	Sometim	100-500	No	Yes	Yes	Yes	No	No	No	Very diffi	Maybe
27-08-2014 11:32	23	Male	Canada	NA	NA	NA	No	No	Never	26-100	No	Yes	Don't knor	No	Don't knor	Don't knor	Don't knor	Don't knor	No
27-08-2014 11:32	31	Male	United	Stu	OH	NA	No	Yes	Sometim	Jun-25	Yes	Yes	Don't knor	No	Don't knor	Don't knor	Don't knor	Don't knor	No
27-08-2014 11:32	29	male	Bulgaria	NA	NA	NA	No	No	Never	100-500	Yes	Yes	Don't knor	Not sure	No	No	Don't knor	Don't knor	No
27-08-2014 11:33	42	female	United	Stu	CA	NA	Yes	Yes	Sometim	26-100	No	No	Yes	Yes	No	No	Don't knor	Somewha	Yes
27-08-2014 11:33	36	Male	United	Stu	CT	NA	Yes	No	Never	500-1000	No	Yes	Don't knor	Not sure	No	Don't knor	Don't knor	Don't knor	No
27-08-2014 11:33	27	Male	Canada	NA	NA	NA	No	No	Never	Jun-25	No	Yes	Don't knor	Not sure	No	Don't knor	Don't knor	Somewha	No
27-08-2014 11:34	29	female	United	Stu	IL	NA	Yes	Yes	Rarely	26-100	No	Yes	Yes	Not sure	No	No	Don't knor	Somewha	No
27-08-2014 11:34	23	Male	United	Kir	NA	NA	No	Yes	Sometim	26-100	Yes	Yes	Don't knor	No	Don't knor	Don't knor	Don't knor	Very easy	Maybe
27-08-2014 11:34	32	Male	United	Stu	TN	NA	No	Yes	Sometim	Jun-25	No	Yes	Yes	Yes	No	Don't knor	Don't knor	Don't knor	Maybe
27-08-2014 11:34	46	male	United	Stu	MD	Yes	Yes	No	Sometim	03-May	Yes	Yes	Yes	Not sure	Yes	Don't knor	Yes	Very easy	No

Fig. 2 Suvery.csv file structure to link with machine learning algorithms

of the supervised learning methodologies [15]. This package consists of Numerical Python in short called as NumPy is a library used to create objects related to multi-dimensional arrays. The next tool or library that is associated with dataset is pandas, which means panel data, which is a data set tool for cleaning, transforming, and analyzing. The next step is to associate with graphical analysis tool or library and matplotlib is used for graphical analysis as a plotting library are utilized as part of execution and associated with Scikit-learn along with seaborn is a data visualization library based on matplotlib [17]. Anaconda is the tool used for data science and machine learning based on it, Jupyter Notebook is used to write the code (Figs. 1 and 2).

The data of csv file (comma separated values) file can be identified and retrieved based on the following commands.

1. data = pd.read_csv('C:\dataset\survey.csv').
2. Print(data)

The different variables available in the dataset of mental health are that of the following (Fig. 3).

The different factors that influence the mental health survey are as shown in the above screen shot, mainly family history, wellness_program, mental_health_consequence, physical health consequence, and finally comparison of mental and physical health. Stress, anxiety, and depression are the common factors that influence the mental health [18] (Fig. 4).

data.dtypes	
Timestamp	object
Age	int64
Gender	object
Country	object
state	object
self_employed	object
family_history	object
treatment	object
work_interfere	object
no_employees	object
remote_work	object
tech_company	object
benefits	object
care_options	object
wellness_program	object
seek_help	object
anonymity	object
leave	object
mental_health_consequence	object
phys_health_consequence	object
coworkers	object
supervisor	object
mental_health_interview	object
phys_health_interview	object
mental_vs_physical	object
obs_consequence	object
comments	object
dtype:	object

Fig. 3 Screenshot of the variables of the Survey.csv

The machine learning algorithms that are used for the supervised learning algorithms that are adopted are logistic regression, K Nearest neighbors' classifier, Decision Tree classifier, and Random forests classifier. We also used an ensemble learning methods such as bagging, boosting, and stacking to improve the performance of the machine learning algorithms (Fig. 5 and Table 1).

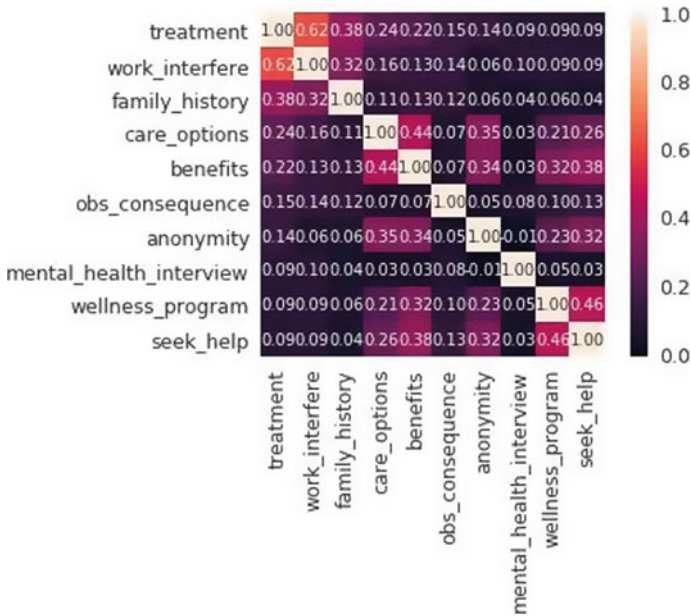


Fig. 4 Graphical analysis of the variables of the Survey.csv

4 Conclusion

The ensemble methodologies are the latest techniques that are used in machine learning approach for better performance and a high rate of accuracy. In experimental setup, we used the Anaconda navigator which includes the open-source Python packages such as NumPy and Pandas, where Pandas is used to extract the dataset that is spambase.csv. We use Jupiter notebook for coding and used scikit-learn. Experimental and graphical analysis is given by using the mental health database that is survey.csv, where the results we received the highest accuracy is of Random Forest algorithm with 81.22, whereas the boosting gave a little bit more accuracy level with the percentage of accuracy 81.75. The boosting, bagging, and stacking are the ensemble learning methods that improves the performance of the machine learning algorithms [8]. Finally, we conclude by informing that the Artificial Intelligence-based Robotic Technology, data mining, and machine learning play a major role in the principles and procedures adoption on datasets for better results and policies [19].

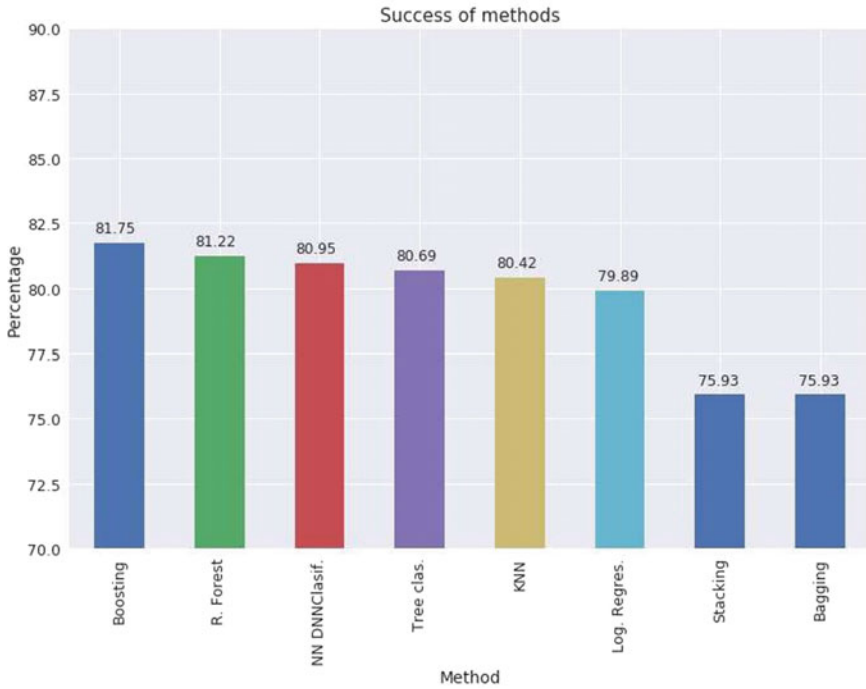


Fig. 5 Graphical analysis of the various machine learning methodologies

Table 1 Classification table of algorithms implemented on the datasets

S. no	Machine learning algorithms	Percentage (Rate) of accuracy
1	Random Forest	81.22
2	K-nearest neighbors algorithm	80.42
3	Decision tree classification	80.69
4	Logistic regression	80.95
5	Bagging	75.93
6	Boosting	81.75
7	Stacking	75.93



References

1. Paul A, George J (2020) Mental health data analysis using cloud. In: 2020 international conference on electrical, communication, and computer engineering (ICECCE), pp 1–4. <https://doi.org/10.1109/ICECCE49384.2020.9179289>
2. Vijaya Chandra J, Challa N, Ali Hussain M (2014) Data and information storage security from advanced persistent attack in cloud computing. *Int J Appl Eng Res* 9(20):7755–7768

3. Jadala VC, Pasupuletti SK, Raju SH, Kavitha S, Sai Bhaba CMH, Sreedhar B (2021) Need of Internet of Things, industrial IoT, industry 4.0 and integration of cloud for industrial revolution. In: 2021 innovations in power and advanced computing technologies (i-PACT), pp 1–5. <https://doi.org/10.1109/i-PACT52855.2021.9696696>
4. Chandra JV, Challa N, Pasupuleti SK (2016) Advanced persistent threat defense system using self-destructive mechanism for cloud security. *IEEE Int Conf Eng Technol (ICETECH)* 2016:7–11. <https://doi.org/10.1109/ICETECH.2016.7569181>
5. Vijaya J, Narasimham C, Sai C, Pasupuleti K (2015) Intelligence based defense system to protect from advanced persistent threat by means of social engineering on social cloud platform. *Indian J Sci Technol* 8(28), October 2015
6. Vijaya Chandra J, Challa N, Pasupuleti SK. Detection of deceptive phishing based on machine learning techniques. In: *Smart technologies in data science and communication. Lecture notes in networks and systems*, vol 105. Springer, Singapore
7. Manekar AK, Pradeepini G (2015) Cloud based big data analytics a review. *Int Conf Comput Intell Commun Netw (CICN)* 2015:785–788. <https://doi.org/10.1109/CICN.2015.160>
8. Thieme A, Belgrave D, Doherty G (2020) Machine learning in mental health. *ACM Trans Comput-Hum Int* 27(5):1–53. <https://doi.org/10.1145/3398069>
9. Xiong Z, Zhang X, Zhang M, Cao B (2020) Predicting features of human mental disorders through methylation profile and machine learning models. In: 2020 2nd international conference on machine learning, big data and business intelligence (MLBDBI), pp 67–75. <https://doi.org/10.1109/MLBDBI51377.2020.00019>
10. Harita U, Kumar VU, Sudarsa D, Krishna GR, Basha CZ, Kumar BSSP (2020) A fundamental study on suicides and rainfall datasets using basic machine learning algorithms. In: 2020 4th international conference on electronics, communication and aerospace technology (ICECA), pp 1239–1243. <https://doi.org/10.1109/ICECA49313.2020.9297440>
11. Pasupuleti SK, Vijaya Chandra J, Challa N (2020) Cross validation of an effective machine learning model on unified data sets to detect and analyse spear phishing attacks. *Studia Rosenthaliana (J Study Res)* 12(6): 63–71, June 2020
12. Chandra JV, Challa N, Pasupuleti SK (2016) A practical approach to Email spam filters to protect data from advanced persistent threat. In: 2016 international conference on circuit, power and computing technologies (ICCPCT), Nagercoil, pp 1–5
13. Saibaba CMH, Waris SF, Raju SH, Sarma V, Jadala VC, Prasad C (2021) Intelligent voice assistant by using OpenCV approach. *Second Int Conf Electron Sustain Commun Syst (ICESC)* 2021:1586–1593. <https://doi.org/10.1109/ICESC51422.2021.9532956>
14. Reddy US, Thota AV, Dharun A (2018) Machine learning techniques for stress prediction in working employees. *IEEE Int Conf Comput Intell Comput Res (ICCIC)* 2018:1–4. <https://doi.org/10.1109/ICCIC.2018.8782395>
15. Chandra JV, Ranjith G, Shanthisri A, Rakshitha R, Mahesh K (2019) A framework for implementing machine learning algorithms using data sets. *Int J Innov Technol Explor Eng (IJITEE)* 8(11):155–160
16. Bangare SL, Pradeepini G, Patil ST (2017) Brain tumour classification using mixed method approach. *Int Conf Inform Commun Embed Syst (ICICES)* 2017:1–4. <https://doi.org/10.1109/ICICES.2017.8070748>
17. Vijaya Chandra J, Challa N, Pasupuletti SK (2019) Machine learning framework to analyze against spear phishing. *Int J Innov Technol Explor Eng (IJITEE)* 8(12), October 2019, ISSN: 2278-3075
18. Ranjith G, Vijayachandra J, Prathusha B, Sagarika P (2016) Design and implementation of a defense system from TCP injection attacks. *Indian J Sci Technol* 9(40):6–11
19. Riek LD (2016) Robotics technology in mental health care. *Artif Intell Behav Mental Health Care* 185–203. <https://doi.org/10.1016/b978-0-12-420248-1.00008-8>

Course Difficulty Estimation Based on Mapping of Bloom's Taxonomy and ABET Criteria



M. Premalatha , G. Suganya , V. Viswanathan ,
and G. Jignesh Chowdary

Abstract The performance of a student is assessed in the present educational system using grades or marks they secure in various levels of examinations. This scoring of marks or grades that a students' scores as part of his/her course depend on diverse parameters. Among those, the difficulty level of a course is identified to be the main parameter considering the history of scoring. Hence, priori computation of the difficulty level for a course may help the both the student and teacher community to decide on the amount of training required for the effective completion of the course. In this work, we have proposed a methodology to estimate the optimal difficulty level of a course by mapping three main components including action words in Bloom's Taxonomy, criteria stated in Accreditation Board for Engineering and Technology (ABET), and the student learning outcomes defined in the course. The difficulty level estimated using the proposed model is validated using the history of grades secured by the students of our university and is found to be promising.

Keywords Bloom's action words · Student learning outcomes · Course difficulty · ABET criteria

1 Introduction

Course difficulty estimation [1, 2] has been a problem for many ages. A single parameter might not be sufficient in estimating the difficulty of a course. Course difficulty is based on various factors. It depends on the student's learning capacity, instructors' capability of delivering the lecture contents, course contents need more cognitive level thinking, courses might need higher-order thinking capability, etc. So obviously it affects the grades of the students. Less difficulty might yield great grades and more difficulty might yield lower grades for a student [2, 3]. Many technics and methodologies such as bloom's taxonomy [4] have been defined to help to assist us in predicting course yet it has not been precise due to having human errors in it as all

M. Premalatha (✉) · G. Suganya · V. Viswanathan · G. Jignesh Chowdary
Vellore Institute of Technology, Chennai, India
e-mail: premalatha.m@vit.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
R. J. Kannan et al. (eds.), *Computer Vision and Machine Intelligence Paradigms for SDGs*, Lecture Notes in Electrical Engineering 967,
https://doi.org/10.1007/978-981-19-7169-3_30

329

technics require human work. Humans cannot provide enough confidence level in the predictions and require more sophisticated mathematical models for prediction. Prediction of course difficulty is vital to understand the quality of a particular course and to study the development of each generation of students. These data can further help us to improve a course based on the development seen in a different generation of students.

1.1 Bloom's Taxonomy

Bloom's taxonomy [4, 4] represents the hierarchical models that are defined to categorize the educational learning objectives into varying levels of complexity and specificity.

These models cover and put all learning objectives into three domains namely cognitive, sensory, and affective domains. The domain with the focus of traditional education is the cognitive domain and this is frequently used to structure the curriculum learning objectives, with appropriate assessments and activities as required. The major categories recorded in the cognitive domain are as follows:

- i. Remembering: Identifying or memorizing the terms, facts, basic concepts, or answers without essentially understanding what they mean to them.
- ii. Comprehension: Demonstrating the understanding about the facts and ideas through different activities including organizing, linking, translating, interpreting, giving descriptions, and stating the main ideas.
- iii. Application: Solve problems occurring in new real-life situations using acquired knowledge, facts, rules, and techniques. Learners may use the prior knowledge to solve problems, and then to identify varying connections and relationships and apply them in new situations.
- iv. Analysis: Examine the problem thereby breaking information into component parts, determining the relationship between them, then identifying the motives or causes, creating inferences, and finding proofs to support generalizations.
- v. Synthesis: Building a base structure from various elements to build a whole component from different parts.
- vi. Evaluation: Defending and presenting opinions by making decisions about information, the validity of thoughts, or the quality of work built on a defined set of criteria.

1.2 ABET Criteria

Accreditation Board for Engineering and Technology (ABET) needs the program to document student outcomes that prepare graduates to attain the corresponding

- (a) an ability to apply knowledge of mathematics, science, and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- (d) an ability to function on multi disciplinary teams
- (e) an ability to identify, formulate, and solve engineering problems
- (f) an understanding of professional and ethical responsibility
- (g) an ability to communicate effectively
- (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- (i) a recognition of the need for, and an ability to engage in life-long learning
- (j) a knowledge of contemporary issues
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.
- (l) An ability to apply mathematical foundations, algorithmic principles and computer science theory in modeling and design of computer-based systems (CBC)
- (m) An ability to apply design and development principles in the construction of software systems (CS)

Fig. 1 ABET Criteria a-m

program educational objectives. ABET Criterion 3 [6] defines student outcomes (a) through (k) plus any additional outcomes that may be articulated by the program to achieve better results.

Figure 1 depicts the student outcomes “a to m” followed by B.Tech CSE program (Curriculum 2014) at Vellore Institute of Technology, Chennai, India.

1.3 Mapping ABET Criteria Students' outcomes with Blooms' Taxonomy

Blooms' Taxonomy is used for evaluating the higher-order thinking and lower order thinking capacity of a student for answering a question during the examination. Bloom's is not only for evaluating the question but it is for classroom teaching tools [7]. Blooms' Taxonomy is mapped with ABET criteria so that a course can be defined with the level of cognitive thinking needed while learning the course. In the proposed method, the difficulty of all courses in the curriculum, by mapping the action words of bloom with the ABET criteria. The sequence of contents discussed in this paper is as given: a detailed literature analysis is elaborated in Sect. 2 followed by Sect. 3 with a discussion on the proposed architecture. The results and discussions are elaborated on Sect. 4 and Sect. 5 presents a conclusion about the work.

2 Literature Survey

The difficulty level of a course can be estimated by considering various parameters. The authors used few parameters like the average grades awarded, rank correlation coefficient (ρ) of a student with the statistical parameters like mean with scaling and cluster analysis [2]. A study on individual item difficulty assessment depicts that the course difficulty depends on the learners, course contents and the scores students have got in the course. The scores are received using apriori algorithm along with the subject matter expert [8]. The impact of course difficulty in students' performance is studied in [9], using the previous student' results and by predicting the results of the current student' using AdaBoost, J48 by AdaBoost, M5P. Based on the learning dependency of an individual, the difficulty of knowledge for different units is ranked with respect to two patterns: subjective difficulty and objective difficulty [10]. Few research works discuss on the way of evaluating the instructors [11]. Parallely, the feedback [12] given by students helps the instructors to improve the performance in examinations.

Course planning [1, 13, 14] is a recommender system in which the students are recommended with the sequence of courses to be taken or registered for each semester. Based on the interest of the students, some courses are recommended [15] in which course recommendations are based on the user profiles from the description of end users' interest, log of browsers, and subscriptions. Learning objects are also recommended sequentially [16] in which the students are recommended with the learning path and advised on how a course contents has to be sequentially learned.

A personalized learning path is recommended to learn the sequence of various learning objects and if the student fails to cater those in the assessment during the learning process, the path will be modified/repared and suggested with new objectives. Blooms' Taxonomy is basically used for evaluating the question paper's cognitive thinking [17, 18]. The performance of the student in the examination based on the difficulty index of a course is assessed through Students' course Outcomes. The final exam papers are evaluated and a difficulty index is identified [19]. Then a cognitive map [is proposed to the research scholars to give them a clear picture of the procedure to start and proceed with their research.

3 Proposed Architecture

In this method, Bloom's Taxonomy is specified with hierarchy and identified with every ABET criteria of student's outcomes is mapped with the action words of Bloom's Taxonomy and arrived at a level of complexity [20]. Figure 2 depicts Bloom's taxonomy hierarchy and Fig. 3 depicts the level of complexity applied to each category of Bloom's taxonomy. By considering Bloom's complexity and based on the action keywords of Bloom's every ABET criteria is mapped with the level of complexity of bloom as specified in Table 1.

Fig. 2 Bloom’s taxonomy [20]

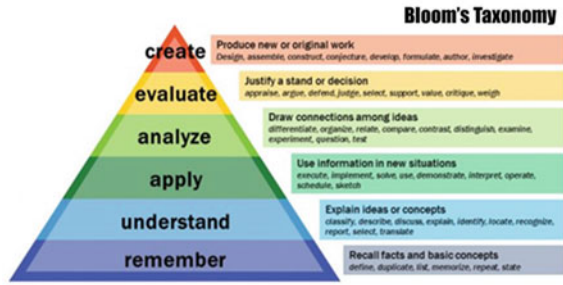


Fig. 3 Level of complexity of Bloom’s taxonomy [21]

Level of Complexity	Cognitive (Knowledge)	Affective (Attitude)	Psychomotor (Skill)
1	Knowledge	Receiving	Imitation
2	Comprehension	Responding	Manipulation
3	Application	Valuing	Precision
4	Analysis	Organizing	Articulation
5	Synthesis	Characterizing by value or value concept	Naturalization
6	Evaluation		

Table 1 Applying the level of complexity for ABET

ABET criteria	Remember (1)	Understand (2)	Apply (3)	Analyze (4)	Evaluate (5)	Create (6)	Total (21)
a	1	2	3				6
b	1	2	3	4	5	6	21
c	1	2	3	4	5	6	21
d	1	2	3				6
e	1	2	3	4	5	6	21
f	1	2					3
g	1	2					3
h	1	2	3				6
i	1	2	3	4	5	6	21
j	1						1
k	1	2	3				6
l	1	2	3	4	5	6	21
m	1	2	3	4	5	6	21

Every criterion of ABET is mapped with the level of complexity of bloom and arrived with the difficulty of the criteria as specified in Table 1. The first two levels of headings alone should be numbered and lower-level headings remain unnumbered. These lower-level headings are formatted as run-in headings.

3.1 Course Difficulty Estimation

If a course is identified with a criterion, the difficulty of a course will be the sum of all difficulty index of the identified ABET criteria as specified in Eq. (1). If a course has a, h, k, l as criteria, difficulty of that course is $6 + 6 + 6 + 21 = 39$. They found difficulty index of a course is cross verified with the mean of the grades scored by the past three generations of students. Figure 4 specifies the procedural steps for estimating the course difficulty.

$$Criteria\ difficulty = \sum_{i=0}^n C_i \tag{1}$$

where

C is the cognitive thinking level assigned for each bloom keyword for a particular criterion.

n is the level of bloom’s taxonomy.

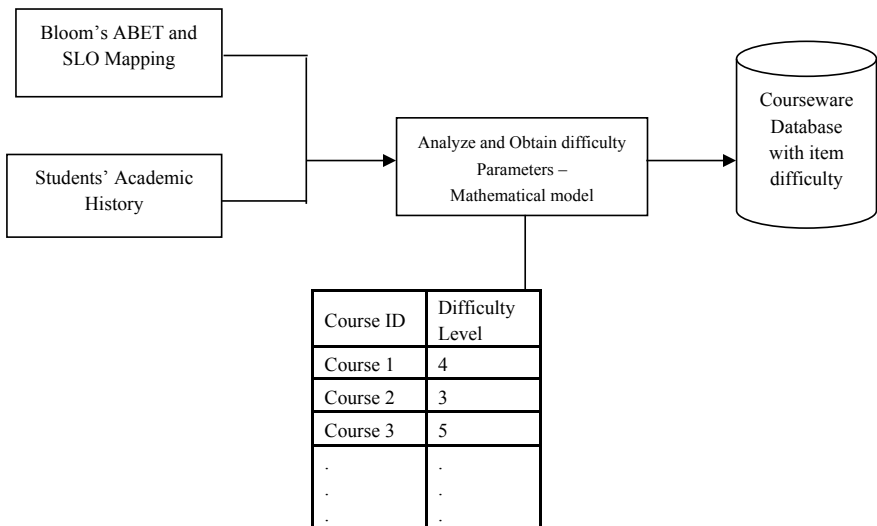


Fig. 4 Course difficulty estimation

Algorithm 1: Course Difficulty Estimation using ABET Student Outcomes and Blooms' Taxonomy

Input: B.Tech. CSE Curriculum with difficulty level as computed using Algorithm 2 and Algorithm 3

Output: Curriculum Courses with a Difficulty level

1. Get the difficulty level of a course found from ABET Blooms' mapping and Student's Academic History using Algorithm 2.
2. Identify the Mean Square Error for each difficulty parameter.
3. Calculate the final difficulty level of a course

Algorithm 2: Difficulty Estimation with Mapping Blooms Rubrics with ABET Mapping

Input: Bloom's taxonomy and ABET Criteria

Output: Curriculum Courses with Difficulty level – Part I

1. Set rubrics for Bloom's Taxonomy hierarchy as 1-6
2. Map Blooms rubrics to ABET Student's Outcomes
3. Mark every mapping of ABET with Bloom's Hierarchy value
4. Repeat it for all ABET criteria
5. Estimate the difficulty rubric for each ABET criteria by summing the Bloom's rubric mapped for each criterion.
6. Level of Higher-order thinking is determined by the difficulty rubric. More the rubric value, more the difficulty level of the criteria is.

Algorithm 3: Difficulty Estimation Students' Academic History

Input: Student's academic history

Output: Courses with Difficulty level – Part II

1. Get student details
2. Get the courses registered by each student with grades for 3 generations
3. Compare the mean for each generation.
4. Average of three generations grades are considered as the difficulty parameter.

Algorithm 4: Validating the Difficulty Estimation

Input: Course Difficulty and student grades

Output: Comparison results

1. Analyze the student's grades for each course
2. If the course difficulty is high and the grades are low for more % of students, we infer that the difficulty of a course has an impact on the students' grades.

In the proposed method, the action words of bloom's are mapped with the ABET criteria, and for each criterion from a-m, the difficulty rubric is calculated using Algorithm 2. Based on the grades of the students, the difficulty of a course is estimated using Algorithm 3. Algorithm 1, compares the difficulty estimated from Algorithms 2 and 3 and set a final difficulty rubric for each course.

4 Results and Discussion

The outcome mapped with each unit of a syllabus is mapped with the bloom's action words and is set with the rubrics as specified in Table 1. From the curriculum, 11 courses mapped with ABET criteria were chosen and applied with the rubric estimated as per Table 1. As per the specifications, the difficulty index of each course is estimated by identifying the number of ABET criteria mapped for each course with the bloom's total score 21 as specified in Table 2. This is part I estimation of the difficulty index as per Algorithm 2. The difficulty index is specified in the Likert scale from 1 to 5 from low to high.

The next part of difficulty estimation is by comparing the student's average grades scored in each course as specified in Algorithm 2. The grades scored by 3 batches of students were considered for identifying the difficulty of a course.

$$DI = 5 - \left(\frac{\text{Class Average}}{100} \right) * 5 \quad (2)$$

The class average is converted into 5 and subtracted with 5 for estimating the Difficulty Index (DI) as specified in Eq. (2). If the class average of a course is 35 out of 100, the difficulty index will be $5 - 1.75$, which is 3.25. Similarly, all the courses are estimated with the difficulty index to the point of 5 as specified in Table 3. The actual DI and the predicted DI are depicted in Fig. 5. Later, the estimated difficulty index with respect to bloom's ABET mappings is compared with the estimated difficulty index with respect to the student's grades and the mean square error is specified in Table 3.

5 Conclusion

The level and kind of training to be given to a course vary from course to course. The difficulty level of the course serves as one major parameter for deciding the number and type of tutoring to be done for the course. Hence, understanding and computing the difficulty level of the course may serve as a boom to the student community which in-turn will directly impact the performance of the student in the examination. Bloom's taxonomy and ABET criteria set a mark in deciding the quality of the system and hence difficulty estimation based on these two will obviously serve as a support

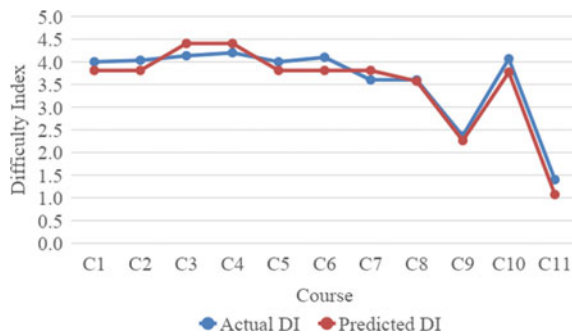
Table 2 Course difficulty estimation wrt. ABET bloom mapping

Course code	a	b	c	d	e	f	g	h	i	j	k	l	m	Total ABET + Bloom Mapping	No. of criteria	Count * 21	Difficulty index
Rubric	6	21	21	6	21	3	3	6	21	1	6	21	21	157			
C1	6	21			21				21		6	21		96	6	126	3.8
C2	6	21			21				21		6	21		96	6	126	3.8
C3		21	21						21		6	21	21	111	6	126	4.4
C4	6	21	21		21				21			21		111	6	126	4.4
C5	6	21							21		6	21	21	96	6	126	3.8
C6	6	21			21				21		6	21		96	6	126	3.8
C7	6	21			21				21		6	21		96	6	126	3.8
C8			21		21					6	6	21		75	5	105	3.6
C9	6						5				6	21		38	4	84	2.3
C10		21			21		5	5	21		6	21		95	6	126	3.8
C11				6		3	3			6				18	4	84	1.1

Table 3 Course difficulty estimation wrt. student’s grades

Course code/SO	Generation 1	Generation 2	Generation 3	Student’s grades (actual difficulty)	ABET bloom (estimated difficulty)	MSE
C1	4.2	3.4	4.4	4.0	3.8	0.2
C2	4.2	4.1	3.8	4.0	3.8	0.2
C3	4.3	3.9	4.2	4.1	4.4	0.3
C4	3.9	4.2	4.5	4.2	4.4	0.2
C5	3.9	3.8	4.3	4.0	3.8	0.2
C6	4.1	4.3	3.9	4.1	3.8	0.3
C7	3.6	3.4	3.8	3.6	3.8	0.2
C8	3.4	3.8	3.6	3.6	3.6	0.0
C9	2.4	2.6	2.1	2.4	2.3	0.1
C10	4.2	3.9	4.1	4.1	3.8	0.3
C11	1.6	1.2	1.4	1.4	1.1	0.3
Average				3.6	3.5	0.2

Fig. 5 Actual DI versus predicted DI



for student and faculty. The estimated difficulty level of the courses is validated using grades of the appropriate courses for students belonging to three different batches. The accuracy which is the ratio of correct mapping of difficulty level to total courses is estimated to be 98%.

References

1. Bassiri D, Schulz EM (2003) Constructing a universal scale of high school course difficulty. *J Educ Meas* 40(2):147–161
2. Mundfrom DJ (1991) Estimating course difficulty. Doctoral dissertation, Iowa State University
3. Joyce A (2016) Course difficulty and its association with student perceptions of teaching and learning—research. *Kentucky J Excell Coll Teach Learn* 14:4

4. Sosniak LA (1994) Bloom's taxonomy. In: Anderson LW (ed). University of Chicago Press
5. Anderson LW, Krathwohl DR (2001) A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives. Longman
6. Criteria for Accrediting Engineering Programs, 2017–2018. <http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2017-2018/#outcomes>. Accessed 9 Dec 2021
7. Athanassiou N, McNett JM, Harvey C (2003) Critical thinking in the management classroom: Bloom's taxonomy as a learning tool. *J Manag Educ* 27(5):533–555
8. Banerjee S, Rao NJ, Ramanathan C (2015, October) Rubrics for assessment item difficulty in engineering courses. In: *Frontiers in education conference (FIE)*. IEEE, pp 1–8
9. Kaur K, Kaur K (2015, September) Analyzing the effect of difficulty level of a course on students performance prediction using data mining. In: *2015 1st international conference on next generation computing technologies (NGCT)*. IEEE, pp 756–761
10. Liu J, Sha S, Zheng Q, Chen L (2010, November) Ranking difficulty of knowledge units based on learning dependency. In: *2010 IEEE 7th international conference on e-business engineering (ICEBE)*. IEEE, pp 77–82
11. Safavi SA, Bakar KA, Tarmizi RA, Alwi NH (2012) What do higher education instructors consider useful regarding student ratings of instruction? Limitations and recommendations. *Procedia Soc Behav Sci* 31:653–657
12. Corelli A (2015) Direct vs. anonymous feedback: teacher behavior in higher education, with focus on technology advances. *Procedia Soc Behav Sci* 195:52–61
13. Pumpuang P, Srivihok A, Praneetpolgrang P, Numprasertchai S (2008, February) Using Bayesian network for planning course registration model for undergraduate students. In: *2nd IEEE international conference on digital ecosystems and technologies, DEST 2008*. IEEE, pp 492–496
14. Xu J, Xing T, Van Der Schaar M (2016) Personalized course sequence recommendations. *IEEE Trans Signal Process* 64(20):5340–5352
15. Pumpuang P, Srivihok A, Praneetpolgrang P (2008, October) Comparisons of classifier algorithms: Bayesian network, C4. 5, decision forest and NBTree for course registration planning model of undergraduate students. In: *IEEE international conference on systems, man and cybernetics, SMC 2008*. IEEE, pp 3647–3651
16. Wang X, Yuan F (2009, June) Course recommendation by improving BM25 to identify students' different levels of interests in courses. In: *International conference on new trends in information and service science, NISS'09*. IEEE, pp 1372–1377
17. Ognjanovic I, Gasevic D, Dawson S (2016) Using institutional data to predict student course selections in higher education. *Internet High Educ* 29:49–62
18. Zainudin S, Ahmad K, Ali NM, Zainal NFA (2012) Determining course outcomes achievement through examination difficulty index measurement. *Procedia Soc Behav Sci* 59:270–276
19. Swart AJ (2010) Evaluation of final examination papers in engineering: a case study using Bloom's taxonomy. *IEEE Trans Educ* 53(2):257–264
20. Armstrong P (2010) Bloom's taxonomy. Vanderbilt University Center for Teaching. <https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/>. Accessed 9 Dec 2021
21. Yang F, Frederick WBL, Rynson WHL (2014) A fine-grained outcome-based learning path model. *IEEE Trans Systems Man Cybern: Syst* 44(2):235–245