Lecture Notes in Electrical Engineering 985

Siddhartha Bhattacharyya
Mario Koeppen
Debashis De
Vincenzo Piuri *Editors*

Intelligent Systems and Human Machine Collaboration

Select Proceedings of ICISHMC 2022

Springer

Lecture Notes in Electrical Engineering

Volume 985

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. ****

Siddhartha Bhattacharyya · Mario Koeppen ·
Debashis De · Vincenzo Piuri
Editors

Intelligent Systems and Human Machine Collaboration

Select Proceedings of ICISHMC 2022

Springer

Editors
Siddhartha Bhattacharyya
Algebra University College
Zagreb, Croatia

Rajnagar Mahavidyalaya
Birbhum, West Bengal, India

Debashis De
Maulana Abul Kalam Azad University
of Technology
Kolkata, West Bengal, India

Mario Koeppen
Kyushu Institute of Technology
Kitakyushu, Fukuoka, Japan

Vincenzo Piuri
University of Milan
Milan, Italy

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-19-8476-1 ISBN 978-981-19-8477-8 (eBook)
https://doi.org/10.1007/978-981-19-8477-8

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

Committee

Advisory Committee

General Chairs

Dr. Siddhartha Bhattacharya, Algebra University College, Zagreb, Croatia and Rajnagar Mahavidyala, West Bengal, India
Dr. Vilas Nitnaware, KCCEMSR, Maharashtra, India

Program Chairs

Dr. Debotosh Bhattacharya, Jadavpur University, West Bengal, India
Dr. Amlan Chakraborti, A. K. Choudhury School of Information Technology, University of Calcutta, West Bengal, India
Dr. Gopakumaran Thampi, Thadomal Engineering College, Maharashtra, India
Dr. Faruk Kazi, VJTI, Maharashtra, India
Dr. Sunil Bhirud, VJTI, Maharashtra, India
Dr. Deven Shah, Thakur College of Engineering and Technology, Maharashtra, India
Dr. Subhash K. Shinde, Lokmanya Tilak College of Engineering, Maharashtra, India

Technical Chairs

Dr. Sudip Kumar Naskar, Jadavpur University, West Bengal, India
Dr. Sourav De, Cooch Behar Government Engineering College, West Bengal, India
Dr. Debanjan Konar, CASUS, Germany

Publication Chairs

Dr. Arundhati Chakraborti, KCCEMSR, Maharashtra, India
Dr. Baban U. Rindhe, KCCEMSR, Maharashtra, India
Dr. Shelley Oberoi, KCCEMSR, Maharashtra, India
Dr. Ravi Prakash, KCCEMSR, Maharashtra, India
Dr. Avishek Ray, KCCEMSR, Maharashtra, India
Dr. Poulami Das, KCCEMSR, Maharashtra, India

Industry Chairs

Dr. M. Chandrashekar Swami, Bharat Dynamics Ltd., Andhra Pradesh, India
Dr. Asima Bhattacharya, Director, TEOCO, West Bengal, India
Dr. Suresh Shan, Mahindra, Maharashtra, India
Mr. Tamaghna Bhattacharya, CarWale, Maharashtra, India
Mr. Abhijit Guha, JP Morgan Chase & Co, Sydney, Australia

Advisory Committee

Dr. Vincenzo Piuri, University of Milan, Australia
Dr. Swapan Bhattacharya, Jadavpur University, West Bengal, India
Dr. Shivaji Bandyopadhyay, NIT Silchar, Assam, India
Dr. Uddhav Bhosale, Swami Ramanand Teerth Marathawada University, Maharashtra, India
Dr. D. Somayajulu, IIIT Kurnool, Andhra Pradesh, India
Dr. Suresh Ukrande, University of Mumbai, Maharashtra, India
Dr. Rajendra Patil, College of Engineering, Maharashtra, India
Dr. Sanjay Pawar, SNDT University, Maharashtra, India
Dr. R. C. Ramola, CFAI University, Dehradun University Campus, Uttarakhand, India
Dr. Avinash Agrawal, SRKN Engineering College, Maharashtra, India
Dr. Ali Yazici, Atilim University, Ankara, Turkey
Dr. Nemkumar Banthia, University of British Columbia, Canada
Dr. D. Bakken, Washington State University, USA
Dr. V. George, Lal Bahadur Shashtri COE, Kerala, India
Dr. S. Sasikumaran, King Khalid University, Saudi Arabia

Technical Program Committee

Dr. Dipankar Das, Jadavpur University, West Bengal, India
Dr. Chintan Mondal, Jadavpur University, West Bengal, India
Dr. Saumya Hegde, National Institute of Technology Karnataka Shuratkal, Karnataka, India
Dr. Nilanjan Dey, JIS University, West Bengal, India
Dr. Richard Jiang, Lancaster University, UK
Dr. Mariofanna Milanova, University of Arkansas, USA
Dr. Leo Mrsic, Algebra University College, Croatia
Dr. Rahul Deb Das, IBM, Germany
Dr. Shakeel Ahmed, King Faisal University, Kingdom of Saudi Arabia
Dr. Jan Platos, VSB Technical University of Ostrava, Czech Republic
Dr. Balachandran Krishnan, CHRIST (Deemed to be University), Karnataka, India
Dr. Ivan Cruz-Aceves, Center for Research in Mathematics (CIMAT), Mexico
Dr. Rajarshi Mahapatra, IIIT Naya Raipur, Chhatisgarh, India
Dr. Xiao-Zhi Gao, Alto University, Finland
Dr. T. V. S. Arun Murthy, Nova College of Engineering and Technology, Andhra Pradesh, India
Dr. Kiritkumar Bhatt, Sardar Vallabhbhai Patel Institute of Technology, Gujarat, India
Dr. Prakash D. Vyavahare, Shri Govindram Seksaria Institute of Technology and Science (SGSITS), Madhya Pradesh, India
Dr. Debdutta Pal, Brainware University, West Bengal, India
Dr. Sandip Rakshit, American University of Nigeria
Dr. Nabajyoti Mazumder, IIIT Allahabad, Uttar Pradesh, India
Dr. Karan Singh, JNU, Delhi, India
Dr. Ratnesh Natoria, Jaypee University, Uttar Pradesh, India
Dr. Abhishek Srivastava, IIT Indore, Madhya Pradesh, India
Dr. Guru Prakash, IIT Indore, Madhya Pradesh, India
Dr. N. Jha, South Asian University, Delhi, India
Dr. Sobhan Sarkar, University of Edinburgh, UK
Dr. Sachin Jain, Oklahoma State University, OK, USA

Dr. Gerd Moeckel, Heidelberg, Germany
Dr. Samarjeet Borah, SMIT, Sikkim, India
Mr. Pabak Indu, Adamas University, West Bengal, India

List of Reviewers

Sourav De, dr.sourav.de79@gmail.com
Sandip Dey, dr.ssandip.dey@gmail.com
Abhishek Basu, idabhishek23@yahoo.com
Indrajit Pan, p.indrajit@gmail.com
Anirban Mukherjee, anirbanm.rcciit@gmail.com
Ashish Mani, mani.ashish@gmail.com
Debabrata Samanta, debabrata.samanta369@gmail.com
Debanjan Konar, konar.debanjan@gmail.com
Tulika Dutta, munai.tulika@gmail.com
Abhishek Gunjan, abhishek.gunjan@res.christuniversity.in
Debarka Mukhopadhayay, debarka.mukhopadhyay@gmail.com
Abhijit Das, ayideep@yahoo.co.in
Pampa Debnath, poonam.4feb@gmail.com
Arpan Deyasi, arpan.deyasi@gmail.com
Soham Sarkar, sarkar.soham@gmail.com
Koyel Chakraborty, koyel.chak88@gmail.com
Kousik Dasgupta, kousik.dasgupta@gmail.com
Koushik Mondal, gemkousk@gmail.com
Rik Das, rikdas78@gmail.com
Jyoti Sekhar Banerjee, tojyoti2001@yahoo.co.in
Hiranmoy Roy, hiru.roy@gmail.com
Soumyadip Dhar, rccsoumya@gmail.com
Soumyajit Goswami, soumyajit.goswami@gmail.com
Goran Klepac, goran@goranklepac.com

About the Conference

- We have received a total of 108 submissions. Out of this, only 20 papers were accepted and presented.
- Out of twenty papers, there were three papers which were authored by authors from foreign countries namely USA, Canada and UK. Rest all the papers were from institutes in different parts of India.
- Totally, there were four technical sessions. In each session, five papers were presented.
- Each session was attended by more than 50 participants.
- There were a total of seven speakers from different well-known academic institutions from abroad as well as various parts of India; also, there were two speakers from industry as well.

Table 1 Details of Keynote Speakers

Serial No.	Name of Speaker	Affiliation
1.	Dr. Rajkumar Buyya	Professor, CLOUDS Lab, School of Computing and Information Systems, The University of Melbourne, Australia
2.	Dr. Amit Konar	Professor, Department of Electronics and Telecommunication Engineering, Jadavpur University
3.	Dr. Koushik Mondal	Principal System Engineer, IIT Dhanbad
4.	Dr. K. M. Bhurchandi	Professor, Department of Electronics and Communication Engineering, VNIT, Nagpur
5.	Mr. Abhishek Patodia	President, CarTrade Tech Pvt. Ltd., Mumbai
6.	Dr. Aparajita Khan	School of Medicine, Stanford University, USA
7.	Mr. Saukarsha Roy	Country Head, Data Science Division. Loreal India Pvt. Ltd.
8.	Dr. Debanjan Konar	Center for Advanced Systems Understanding, Helmholtz-ZentrumDresden-Rossendorf (HZDR), Germany
9.	Dr. Rajarshi Mahapatra	Dean(Academics), IIIT-Naya, Raipur

Table 2 Details of Session Chairs

Serial No	Name of Session Chair	Affiliation
1.	Dr. Shelly Oberoi	Associate Professor and Head, Department of Humanities and Applied Sciences, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
2.	Dr. Ratnesh Litoria	Associate Professor, Department of Electronics and Computer Engineering, Medi-Caps University, Indore
3.	Dr. Rajiv Iyer	Associate Professor and Head, Department of Electronics and Telecommunication Engineering, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
4.	Dr. Avishek Ray	Associate Professor, Department of Electronics and Telecommunication Engineering, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
5.	Dr. Ravi Prakash	Associate Professor, Department of Computer Engineering, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
6.	Dr. Baban U. Rindhe	Professor, Department of Electronics and Telecommunication Engineering, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra and Member-BoS in EXTC, University of Mumbai
7.	Prof. Yogesh Karunakar	Assistant Professor, Department of Electronics and Telecommunication Engineering, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane and Head E-Cell, KCCEMSR
8.	Dr. Arundhati Chakrabarti	Vice Principal and IQAC Head, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
9.	Prof. Mandar Ganjapurkar	Head, Department of Computer Engineering, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
10.	Dr. Kiran Bhandari	Professor, Department of Information Technology, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
11.	Prof. Amarja Adgaonkar	Head, Department of Information Technology, K. C. College of Engineering and Management Studies and Research (KCCEMSR), Thane, Maharashtra
12.	Prof. Pabak Indu	Assistant Professor, Department of Computer Science and Engineering, Adamas University, West Bengal, India

About This Book

Excelsior Education Society's K. C. College of Engineering and Management Studies and Research, Thane, for the past two decades, is on the mission of imparting technical education in the Thane region.

K. C. College of Engineering and Management Studies and Research, a Punjabi Linguistic Minority Institute, is NAAC accredited and two of its courses are NBA accredited.

The college is affiliated to the University of Mumbai and is approved by AICTE. The institute has an environment conducive to innovation and research and provides training in cutting-edge technology to its students to keep them ready for the global world.

With a vision of holistic education, KCCEMSR provides its students with a wide range of activities from curricular to extracurricular, social immersion programs and universal Human Value Sessions.

Contents

About the Editors

Dr. Siddhartha Bhattacharyya [FRSA, FIET (UK), FIEI, FIETE, LFOSI, SMIEEE, SMACM, SMAAIA, SMIETI, LMCSI, LMISTE] is currently the Principal of Rajnagar Mahavidyalaya, Birbhum, India. Prior to this, he was a Professor at CHRIST (Deemed to be University), Bangalore, India. He also served as the Principal of RCC Institute of Information Technology, Kolkata, India. He has served VSB Technical University of Ostrava, Czech Republic as a Senior Research Scientist. He is the recipient of several coveted national and international awards. He received the Honorary Doctorate Award (D. Litt.) from The University of South America and the SEARCC International Digital Award ICT Educator of the Year in 2017. He was appointed as the ACM Distinguished Speaker for the tenure 2018–2020. He has been appointed as the IEEE Computer Society Distinguished Visitor for the tenure 2021–2023. He is a co-author of 6 books and the co-editor of 94 books and has more than 400 research publications in international journals and conference proceedings to his credit.

Mario Koeppen studied Physics at the Humboldt University of Berlin and received his Master's degree in Solid-State Physics in 1991. He has published over 150 peer-reviewed papers in conference proceedings, journals, and books and was active in the organization of various conferences as a chair or a member of the program committee, including the WSC online conference series on Soft Computing in Industrial Applications, and the HIS conference series on Hybrid Intelligent Systems. He is a founding member of the World Federation of Soft Computing and since 2016 the editor-in-chief of its Elsevier *Applied Soft Computing* journal. In 2006, he became a JSPS fellow at the Kyushu Institute of Technology in Japan, in 2008 a professor at the Network Design and Research Center (NDRC), and in 2013 a professor at the Graduate School of Creative Informatics of the Kyushu Institute of Technology, where he is conducting now research in the fields of soft computing, especially for multi-objective and relational optimization, digital convergence, and human-centered computing.

Debashis De is a Professor at the Department of Computer Science and Engineering at Maulana Abul Kalam Azad University of Technology, West Bengal, India. He is Senior Member-IEEE, Fellow IETE, and Life member CSI. He was awarded the prestigious Boyscast Fellowship by the Department of Science and Technology, Government of India for a postdoc in Scotland, UK. He received the Endeavour Fellowship Award from 2008 to 2009 by DEST Australia to work at the University of Western Australia. He received the Young Scientist award both in 2005 at New Delhi and in 2011 in Istanbul, Turkey, from the International Union of Radio Science, Belgium. He established the Center of Mobile cloud computing. He is the Vice-chair of Dew Computing STC of IEEE Computer Society. His research interest is cloud, IoT and Quantum Computing.

Vincenzo Piuri is a full professor at the University of Milan, Italy (since 2000), where he was also the department chair (2007–2012). He received his M.S. and Ph.D. in Computer Engineering from Politecnico di Milano, Italy. His research and industrial application interests are artificial intelligence, computational intelligence, intelligent systems, pattern analysis and recognition, machine learning, signal and image processing, biometrics, intelligent measurement systems, industrial applications, distributed processing systems, Internet of things, cloud computing, fault tolerance, application-specific digital processing architectures, and arithmetic architectures. He published innovative results in over 400 papers in international journals, international conference proceedings, books, and chapters.

Landmark Identification from Low-Resolution Real-Time Image for Pose Estimation

Rajib Sarkar, Siddhartha Bhattacharyya, Debashis De, and Asit K. Datta

Abstract The study of human posture estimation has produced an excellent illustration of the present state of human and computer vision. Various experts from across the world are now performing considerable studies on this topic. Human action recognition and posture are key criteria for preserving consistency inside real-time backdrops and objects. In order to comprehend human behavior, key points must be identified. One can better comprehend future functions by evaluating mechanical models of the human body. This article proposes a new method to identify and anticipate human action key points. A full description of the recent methods related to human posture estimation is also illustrated with reference to a benchmark database. When discussing the suggested strategy's outcomes and conversations, it is clear that the proposed approach has superior performance and works considerably better than the previous approaches. It is predicted that the proposed technique would provide a new dimension to humanistic recognition observation research.

Keywords Human action recognition · Posture assumption · Key point identification · Human behavior · Human–computer interaction

R. Sarkar
Department of Computer Science and Technology, Nibedita Polytechnic, Kolkata, India

S. Bhattacharyya (✉)
Algebra University College, Zagreb, Croatia
e-mail: dr.siddhartha.bhattacharyya@gmail.com

Rajnagar Mahavidyalaya, Birbhum, India

D. De
Maulana Abul Kalam Azad University of Technology, West Bengal, Kolkata, India

A. K. Datta
University of Calcutta, Kolkata, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_1

1 Introduction

Pose detection is a hot topic of research. In the realm of computer vision, hundreds of research articles and models have been published in an attempt to tackle the challenges involved in pose detection. Pose estimations are appealing to a large number of machine learning enthusiasts due to their vast range of applications and utility.

It is a natural instinct of human beings to perform physical activities supported by limbs. Guessing human posture is one of the most common problems in computer vision. This study has been going on for the last few years and more, the most important reason being that a lot of applications can benefit from this technology. For example, human posture assumptions make high-level reasoning decisions in the context of recognition of the joint play and activity of the computer. Clinical experts can make useful observations from human animations. Recently it has been observed that the method of action recognition and posture estimation is complex to use in some appearance models [1–3], and by reviewing the algorithms, different models can be estimated through different pieces of training. Depending on the importance of the performance of these methods, the appearance, strength, clarity, and image of fresh human clothing obtained from natural language-based training images can now be explained. Despite many years of research, posture estimation remains a very difficult task. Human behavior estimates are usually made for calculations based on some underlying assumptions. The article presents a novel methodology for identifying and anticipating actions. This paper looks at one such use of pose detection and estimation with a COCO pre-trained mechanical model supplemented by a proposed key point identification approach.

The article is organized as follows. With the backdrop provided in the Introduction, a summary of past works in this direction is provided in Sect. 2. Section 3 provides a description of the database used in this study. The proposed methodology is presented in Sect. 4. Section 5 presents the experimental results. Finally, Sect. 6 concludes the paper.

2 Related Works

Pose estimation is a computer vision technique for tracking the movements of a person or an object. This is normally accomplished by locating spots critical to the subjects that can be compared so that different actions, postures, and conclusions can be taken under consideration based on these essential points. In the fields of augmented reality, animation, gaming, and robotics, several models exist for pose estimation which includes Open pose, Pose net, Blaze pose, Deep Pose, Dense pose, and Deep cut [26].

Interactions between humans and computers are a crucial part of real-world applications. Action detection from a video is an integral component of video surveillance systems. According to a review of human activity recognition methods [4], most of the strategies are characterized based on two questions: "what actions" and "what happens for the actions". The majority of human activity recognition techniques apply to a wide range of human activities, including group actions, behaviors, events, gestures, atomic action, human-to-object, human-to-human, and human-to-human interactions. The first step involved in this process is to remove the background, followed by human tracking, human activity, and finally object detection. In [5], human action recognition was discovered using sensor-based techniques. Videos for human action detection are usually captured using sensor-based cameras, which has been a focus of automatic computer vision research.

Media pipe [6] is a cross-platform open-source tool for constructing multimodal machine learning pipelines. It can be used to implement advanced models such as human face detection, multi-hand tracking, hair segmentation, item detection and tracking, and more [6].

Blazing pose detector technology [21] is based on the COCO topology, which has 17 important points. The blaze pose detector predicts 33 essential points in the human body including the torso, arms, legs, and face. For successful applications of domain-specific pose estimation models, such as for hands, face, and feet, more critical elements must be included. Each critical point, as well as the visibility score, is predicted with three degrees of freedom [7]. The blaze pose is a sub-millisecond model that is more accurate than most of the existing models and can be employed in real-time applications to achieve a balance of speed and accuracy. The model is available in two versions, viz., Blaze pose lite and Blaze pose completely.

Various types of deep learning algorithms have been applied on single perspective datasets during the last few years. This is because single perspective human action recognition not only forms the foundation and uses large-scale datasets, but also due to the fact that the architecture created for single views can be directly expanded to multiple viewpoints by constructing multiple networks.

CNNs have also become a prominent deep learning approach in the human action detection space not only for their ability to learn visual patterns directly from image pixels without any pre-processing. A two-step neural network-based deep learning approach was introduced by Baccouche et al. [8]. The first stage uses CNNs to automatically learn spatio-temporal characteristics, followed by a Recurrent Neural Network (RNN) to classify the sequence.

LSTMs are techniques that use memory blocks to replace traditional network units. The gate neurons of the LSTMs control whether the value should be remembered, forgotten, or outputted. Previously, it has been employed to identify speech and handwriting. To solve the issue of traditional LSTMs, Veeriah et al. [9] proposed a new gating method that highlights the shift in information gain induced by prominent movements between subsequent frames. The LSTM model, also referred to as the differential RNN, can automatically detect actions from a single view or a deep dataset.

Shu et al. [10] created a unique model based on SNNs, which is a various leveled structure of feed-forward spiking neural networks that models two visual cortical areas, viz., the primary visual cortex (VI) and the middle temporal area (MT), both of which are neurobiologically dedicated to motion processing.

A posed-based CNN [11] descriptor was developed based on human postures for the purpose of action recognition. The input data was been split into five parts. Two types of frames were retrieved from the movie for each patch with RGB and flow frames. After the aggregation and normalization phases, the P-CNN features were created by both the frames and processed in the CNN.

Deep learning approaches have recently been presented. They have been frequently employed in fields including speech recognition, language processing, and recommendation systems, among others. Since hierarchical statistical approaches offer numerous benefits, such as raw data input, self-learned features, and high-level or complicated action identification, deep learning techniques have sparked a lot of attention. Researchers are able to develop a real-time, adaptable, and high-performing recognition system based on these benefits.

3 Database Description

On July 6, 2021, the FAIR-Habitat 2.0 [25] database was released. This has been used to teach robots how to navigate in three-dimensional virtual environments. This database is able to interact with the items in the same manner as a real kitchen, dining room, and other commonly used rooms. The collection provides information about each item in the 3D scene, such as its size and constant resistance as well as whether or not the item has any parts that may open or close. Habitat 2.0 has 111 distinct living space outlines and 92 items. On September 24, 2019, Audi released the A2D2 [13] dataset for autonomous driving. This new dataset, which is the latest in a long line of company dataset releases, is meant to aid university researchers and businesses working in the field of autonomous driving. More than 40,000 classified camera frames, as well as 2D semantic divisions, 3D point clouds, 3D bounding boxes, and vehicle bus data, are included in the Autonomous Driving Dataset. According to the dataset description, only a portion of the dataset with the 3D bounding boxes covers four distinct annotations. Audi added an extra 390,000 unlabeled frames to the sample as a partial explanation.

YouTube-8 M [14], a data set from Google AI, was released on June 30, 2019. This dataset is made up of data extracted from YouTube videos so that time-based localization and video division structure can offer up a slew of new possibilities. The TrackingNet [15] dataset, which was released in 2018, has a total of 30,132 (train) and 511 (test) movies, indicating that object tracking in the wild is still a work in progress. Current object trackers perform admirably well on well-known datasets, but these datasets are insignificant in comparison to the problems of real-world human action monitoring. However, there is still a need for a dedicated large-scale dataset to train deep trackers. The first advanced frame rate video dataset is the first long-scale dataset TrackingNet for object tracking in the field, at the level of MOT17 for multiple object tracking and required for speed. The KINETICS-600 [16] dataset was

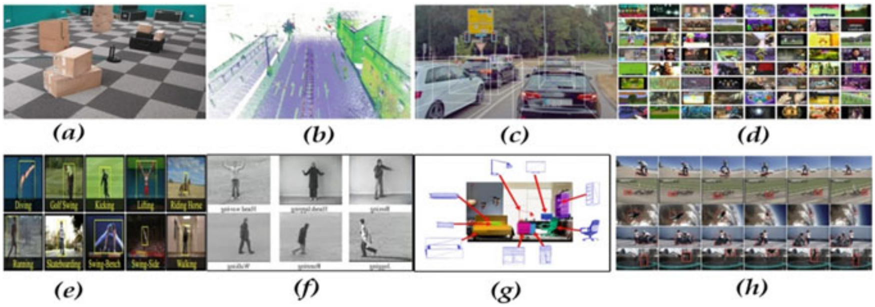


Fig. 1 Data base sample image, **a** FAIR-Habitat 2.0, **b** Audi Released Autonomous Driving Dataset A2D2, **c** Moments in Time, **d** YouTube-8 M, **e** UCF Sports, **f** KTH [18], **g** ObjectNet3D, **h** Moments in Time

introduced in 2021. It contains around 500,000 video segments. This includes at least 600 video clips and 600 human action lessons. Google's DeepMind team has access to all the activity classes in the dataset. However, in 2021, a new KINETICS-700 database with 700 human activity video clips was released. A large-scale high-quality dataset of YouTube URLs has been produced to develop replicas for human action identified. Every clip in Kinetics-600 is derived from a single YouTube video with a runtime of about 10 s and is assigned to a specific class. The clips have gone through several human action circles. Moments in Time [17] is a large-scale dataset that was introduced in 2018 with a total amount of movies in 1,000,000 and 399 human activity categorizations. The MIT-IBM Watson, AI Lab was responsible for this release. A million branded 3-s videos have been distributed. This database isn't just for footage of human acts. People, animals, artifacts, and natural wonders all fall within this category. It also portrays the spirit of a tumultuous situation. There is a significant intra-class difference among the groups in the sample. People, opening doors, gates, drawers, curtains, gifts, and animals are among the actions depicted in the video clips. Figure 1 provides some glimpses of the standard databases.

Action identification and prediction focus on high-level video characteristics rather than identifying action primitives that transform fundamental physical properties. The something-something dataset [19], which examines human-object interactions, is one such example. This dataset, for example, includes labels or textual image templates such as "Dropping [something] into [something]" to label interactions between humans and things, as well as an item and an object. This enables one to develop models that can understand physical elements of the environment, such as human activities, item interactions, spatial relationships, and so on. Databases come in a variety of shapes and sizes and are separated into distinct parts depending on the data type. Table 1 illustrates some well-known databases along with the number of constituent videos and video categories.

Table 1 List of datasets [22]

Datasets	Year	No of data (Videos)	Type
KTH	2004	599	RGB
Weizmann	2005	90	RGB
INRIA XMAS	2006	390	RGB
IXMAS	2006	1148	RGB
UCF Sports	2008	150	RGB
Hollywood	2008	–	RGB
Hollywood2	2009	3,669	RGB
UCF 11	2009	1,100	RGB
CA	2009	44	RGB
MSR-I	2009	63	RGB
MSR-II	2010	54	RGB
MHAV	2010	238	RGB
UT-I	2010	60	RGB
TV-I	2010	300	RGB
MSR-A	2010	567	RGB-D
Olympic	2010	783	EGB
HMDB51	2011	7,000	RGB
CAD-60	2011	60	RGB-D
BIT-I	2012	400	RGB
LIRIS	2012	828	RGB
MSRDA	2012	320	RGB-D
UCF50	2012	50	RGB
UCF101	2012	13,320	RGB
MSR-G	2012	336	RGB-D
UTKinect-A	2012	10	RGB-D
MSRAP	2013	360	RGB-D
Sports-1 M	2014	1,133,158	RGB
3D Online	2014	567	RGB-D
FCVID	2015	91,233	RGB
ActivityNet	2015	28,000	RGB
YouTube-8 M	2016	8,000,000	RGB
Charades	2016	9,848	RGB
ObjectNet3D	2016	90,127	RGB-D
NEU-UB	2017	600	RGB
Kinetics	2017	500,000	RGB-D
AVA	2017	57,600	RGB

(continued)

Table 1 (continued)

Datasets	Year	No of data (Videos)	Type
20BN-Something-Something	2017	108,499	RGB
SLAC	2017	520,000	RGB
MOT17	2017	21(train) + 21 (test)	RGB
Moments in Time	2018	1,000,000	RGB
KINETICS-600	2018	500,000	RGB
TrackingNet	2018	30,132 (train) + 511 (test)	RGB-D
YouTube-8 M	2019	8 M Video	RGB
Audi Released Autonomous Driving Dataset A2D2	2019	40 000	RGB-D
Kinetics-700	2020	700	RGB-D
FAIR-Habitat 2.0	2021	More than one billion	RGB

4 Proposed Methodology

Before discussing the proposed method, it is better to understand the method of evaluating the nature of human motion. It is possible to identify the future motion of a human being by comparing the movement of his limbs with the movement of a mechanical model. The proposed method uses the COCO pre-training mechanical model [21], which is found to be able to understand the finest activities from a low-resolution image.

A video camera is used for the acquisition of the input images from both the indoor and outdoor image scenes. The standard thresholding techniques use fixed and uniform thresholding strategies as they presume that the image intensity information content is homogeneous. The multilevel sigmoidal (MUSIG) activation function [20] however, resorts to some context-based thresholding in order to reflect the image information heterogeneity. Hence, the MUSIG activation function is capable of producing multilevel thresholded outputs corresponding to the multilevel image intensity information. The MUSIG activation function [20] is used in this work for the purpose of object detection by means of identifying the objects (human body) from the acquired image scenes.

After identification of the objects from the image scene, a boundary is created around the objects using the boundary box algorithm [27]. Automatic threshold calculations have been performed on the selected objects and key points of human limb models which have been identified by applying the COCO pre-training mechanical model [21]. This enables real-time applications of the proposed approach.

Figure 2 shows a step-by-step method for estimating human postures from low-resolution color images using the proposed approach.

Thresholding is a way for creating a meaningful representation of image segmentation. The foreground value can be changed depending on the pixel value, and the background value may be changed based on everything else. It is a tough proposition

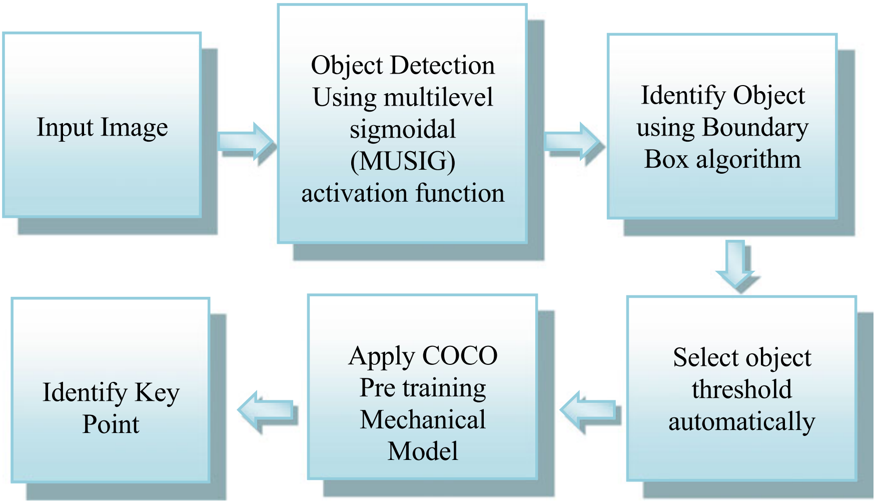

Input Image
Object Detection Using multilevel sigmoidal (MUSIG) activation function
Identify Object using Boundary Box algorithm
Select object threshold automatically
Apply COCO Pre training Mechanical Model
Identify Key Point

Fig. 2 Representation of the key point identification process by the proposed approach

to work on the existence of the primary item based on the light and color of the backdrop of images. As a result, the thresholding technique is used to identify the principal item in all of the test images, including the foreground and background. A global threshold of all the pixels is used in traditional threshold operations. However, an adaptive threshold is a better representation of a dynamic threshold that allows changing the position of the light shift in the images. Shadows can then be used to determine whether or not an activity is intended for light.

The procedure of picking an item from an image employed in this work is discussed. Image segmentation is used to identify the object within the image. Image segmentation is a classification technique based on a precise assessment of the central attributes of various national object areas and other objects. The split of an image allows for a more accurate description and comprehension of an image. The multilevel sigmoidal (MUSIG) activation function [20] is used to examine the single area properties in an image, where the input image estimates the information and content of the real-life image usually displayed as a variation of a bunch of pixels.

Based on this image pixel, the MUSIG activation function [20] selects the thresholding parameter and affects the object and its surrounding actions. The boundary line is used to distinguish items using the boundary box technique [27]. Figure 3 illustrates the proposed strategy of automatic thresholding for object detection using the MUSIG activation function [20] as detailed in Algorithm 1, while Fig. 4 shows the object and border detection without using the MUSIG activation function [20].

It is evident from Fig. 3 that the object border is detected well by using the MUSIG activation function [20], whereas it becomes difficult to detect the object border without using the MUSIG activation function [20] as is evident from Fig. 4.

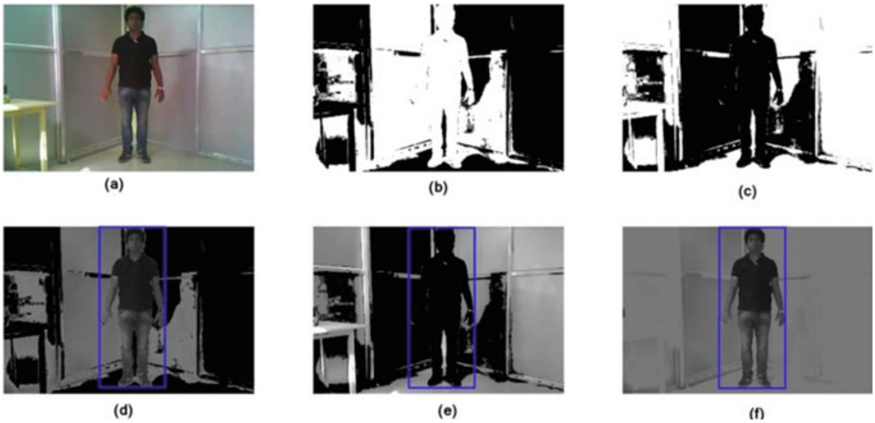


Fig. 3 Object and border detection using multilevel sigmoidal (MUSIG) activation

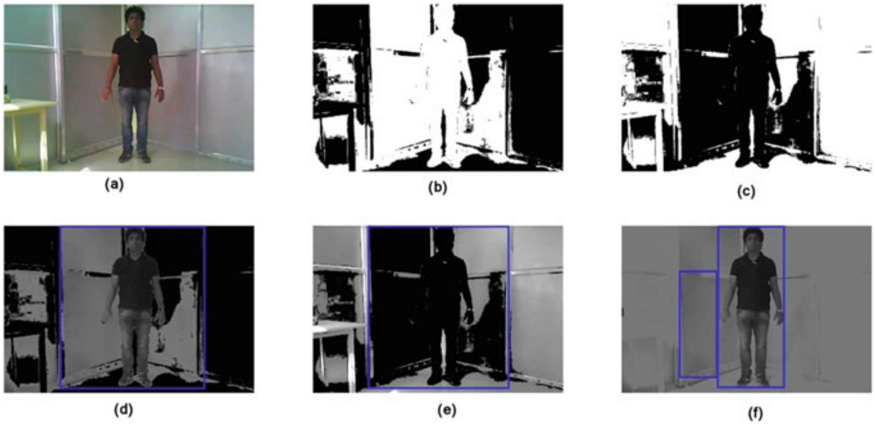


Fig. 4 Object and border detection without using multilevel sigmoidal (MUSIG) activation

Algorithm 1: Automatic Thresholding

Input: RGB image I of size $m\ X\ n$

Output: Automatic threshold.

Step 1: Select an initial estimate of the threshold T. A good initial value is the average intensity of the image *I*.

Step 2: Convert RGB image *I* to a Gray Scale Image.

Step 3: Calculate the mean gray values, μ_1 and μ_2 of the partitions, R_1, R_2

Step 4: Partition the image into two groups, R_1, R_2 using the threshold *T*.

Step 5: Select a new threshold: $T = \frac{1}{2}(\mu_1 + \mu_2)$

Step 6: Repeat steps 2 to 4 until the mean values, μ_1 and μ_2 in successive iterations do not change.

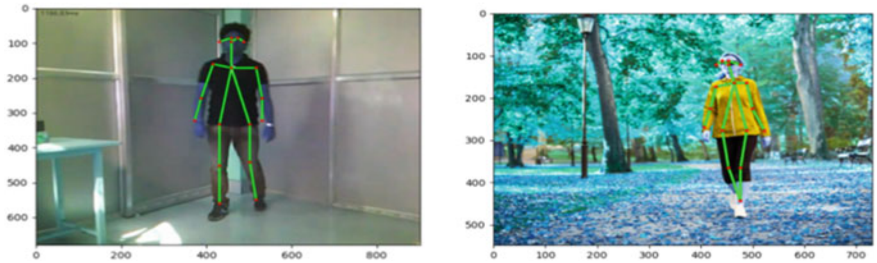


Fig. 5 Representation of important human body key points

Pose estimation is a typical problem in computer vision that involves detecting the origin of an object's location. This generally entails determining the object's location at a critical time. When identifying facial emotions, for example, it looks for human face landmarks. The importance of identifying and stabilizing the key organs of the body, such as the ankle, knee, right elbow of the right hand, and so on, is discussed here. Figure 5 shows a sample solution for identifying significant spots in the body.

The COCO model [21] is a mechanical model by which the computer can identify human activities. This mechanical model identifies the human skeletal ganglia with some numbers from 0 to 32. For example, 0 means the nose, 1 means the right eye inner, 2 means the right eye, 23 means the right hip, etc. as shown in Fig. 6 [21].

The basic idea behind evaluating keypoint detection is to employ the same evaluation metrics that are used for object detection which include the average precision (*AP*), and average recall (*AR*) and their variants. A similarity measure between the ground truth and the anticipated items lies in the foundation of these measures. The Intersection over Union (*IoU*) [24] is used as a similarity metric in the case of object detection (for both boxes and segments). Thresholding of the *IoU* allows to compute precision-recall curves by defining matches between the ground truth and the anticipated items. Thus, a similarity measure analogous to *AP/AR* is created for keypoint

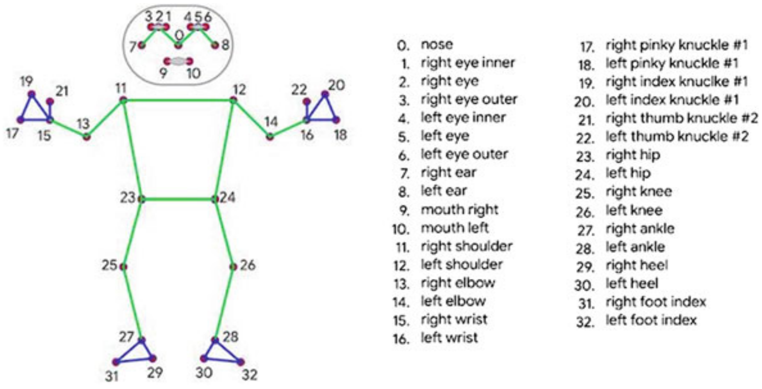


Fig. 6 Representation of the COCO mechanical model key points [21]

detection. However, in order to accomplish this, an object key point similarity (*OKPS*) needs to be introduced, which functions similarly to *IoU*. For each object, *OKPS* ground truth key points are specified as $[X_1, Y_1, V_1 \ldots\ldots\ldots X_K, Y_K, V_k]$ where X, Y are the key point positions and V is a visibility flag defined as $V = 0$: not labeled, $V = 1$: labeled but not visible, and $V = 2$: labeled and visible for each item. In addition, each ground truth object has a scale s, which is defined as the square root of the object segment area.

The keypoint detector must output keypoint locations and object-level confidence for each object. The ground truth and predicted key points for an item should have the same form: $\left[X_1, Y_1, V_{1,} \ldots\ldots\ldots, X_{K,} Y_{K,} V_K\right]$. The keypoint detector is not required to predict per-keypoint visibilities, hence the detector's predicted V_i are not currently employed during evaluation. The object keypoint similarity can be defined as

$$\text{OKPS} = \sum\nolimits_i \left[exp\left(- d_i^2/2s^2k_i^2\right)\delta(V_i > \theta)\right] / \sum_i [\delta(V_i > \theta)] \tag{1}$$

where, d_i are the Euclidean distances between each associated ground truth and detected keypoint, and V_i represents the ground truth's visibility flags (the detector's predicted V_i are ignored). d_i are passed through a normalized Gaussian with standard deviation SK_i to compute *OKPS*, where S is the object scale and K_i is a per-keypoint constant that controls falloff. This generates a keypoint similarity that ranges from 0 to 1 for each keypoint. These commonalities are averaged over all the key points that have been labeled (key points with $V_i > 0$). *OKPS* is unaffected by the predicted key points that are not labeled ($V_i = 0$). $OKPS = 1$ for perfect forecasts, and $OKPS \sim 0$ for those predictions where all the key points are off by more than a few standard deviations. It may be noted that *IoU* and *OKPS* are similar. It can use the object key points to calculate AP and AR, as well as the *IoU* to compute equivalent metrics for box/segment detection.

5 Results and Discussions

This proposed approach has been tested on the KTH human action dataset3 [18] where users are shown in 600 videos (160 × 120) with 25 actors performing six actions in four different scenarios. An example for each of these actions is shown in Fig. 7. It has been observed that the key points of the human pose are very well-identified by applying them on similar low-resolution images.

Approximately 96.2% of key points have been identified with the proposed approach. A comparative study on key point identification between the proposed approach and the open-source bottom-up pose estimation [24] along with other state-of-art methods are shown in Table 2. It is evident from Table 2 that the ability of

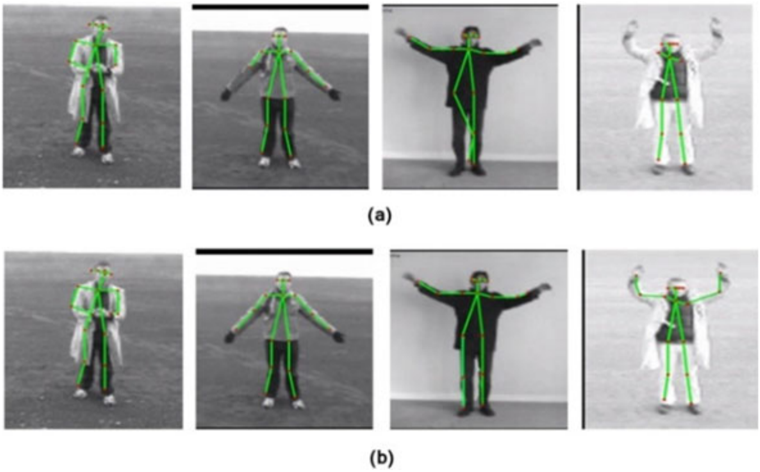


Fig. 7 Comparison between **a** bottom-up pose estimation without using MUSIG [20], **b** proposed approach without using MUSIG [20]

Table 2 Comparison between the proposed approach and other methods [24]

Methods	Keypoint detection %
Pose parsing by	59.14
Without pose NMS	63.92
Without pose completion	64.34
Bounding box constraint	66.03
Bottom-up pose estimation	95.04
Proposed approach	**96.20**

identification by the proposed method is much better compared to the other techniques [indicated by the **boldfaced** value]. Some results of key point identification are shown in Figs. 7, 8, and 9 for the sake of understanding.

When the proposed approach is applied to real-time video frames, it is seen that it is possible to identify the points quite well. The proposed method has been able to better identify the description of real-time images by identifying the key points successfully as shown in Figs. 10 and 11.

A graphical comparison between the bottom-up approach [24] and the proposed approach is shown in Fig. 12. The COCO mechanical model is allocated a total of 32 critical points, as shown in Fig. 6. The *X*-axis is used for the key points, and the *Y*-axis is used as an identifying mark depending on the key points. The suggested technique in Fig. 12b is found to be able to find a greater number of key points.

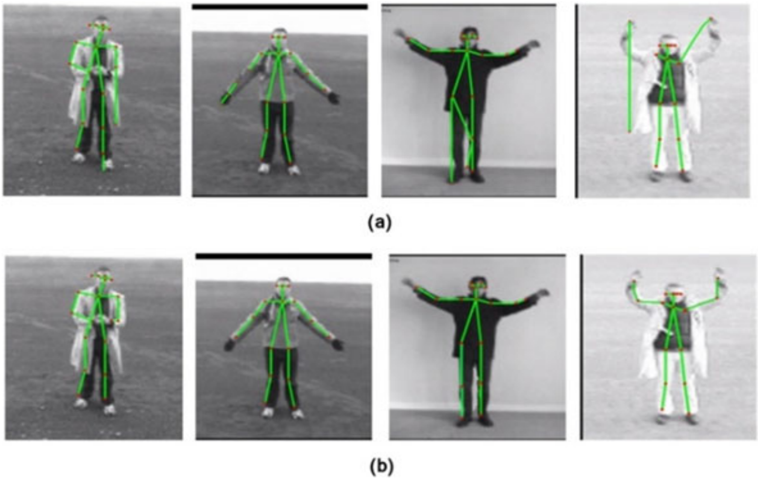


Fig. 8 Comparison between **a** bottom-up without using MUSIG [20], **b** proposed approach using MUSIG [20]

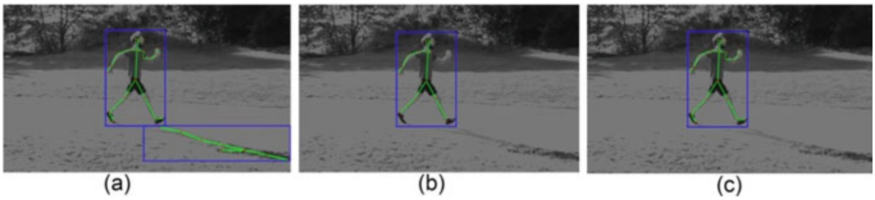


Fig. 9 **a** Proposed approach without using MUSIG [20], **b** Bottom-up pose estimation using MUSIG [20], **c** proposed approach using MUSIG [20]

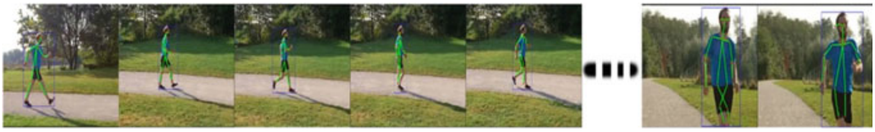


Fig. 10 Representation of a real-time video frame using the proposed approach

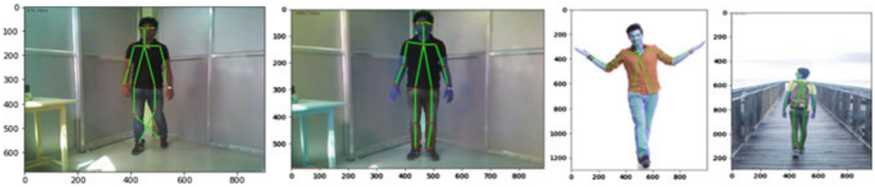


Fig. 11 Representation of keypoint identification by the proposed approach in different environments than the bottom-up approach [24] (shown in Fig. 12a).

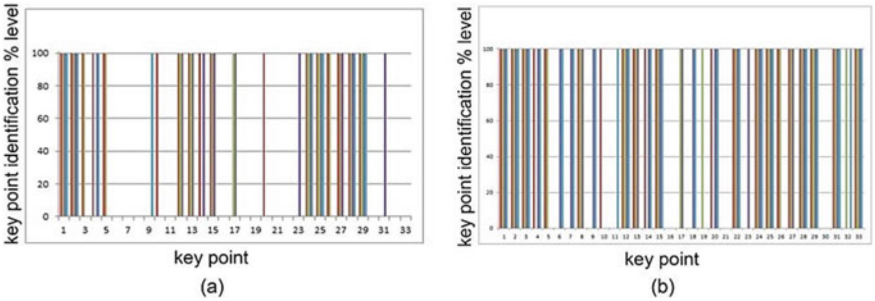


Fig. 12 Graphical representation of key point identification, **a** bottom-up pose estimation and **b** proposed approach

6 Conclusion

In this study, a novel multi-dimensional workspace-based technique for multi-person posture estimation has been presented. The primary concept behind the proposed technique is to learn both the confidence maps of joints and the connection links between the joints using a residual network followed by tracking the posture using a body bounding box for real-time human pose estimation keypoint identification. It shows efficiency in pinpointing critical points using the visual intensity and color information of a low-resolution image. Due to a lesser inference time, it is suitable for real-time operations. The suggested approach may also be used to identify a feature from an image for further estimation. The results reveal that the proposed approach performs exceptionally well in static and runtime situations. The proposed technique is expected to add a new dimension to humanistic recognition observation research.

References

1. Andriluka M, Roth S, Schiele B (2009) Pictorial structures revisited: people detection and articulated pose estimation. In: Proceedings of 2009 24th international conference on pattern recognition (ICPR 2009)
2. Duchi J, Hazan E, Singer Y (2010) Adaptive subgradient methods for online learning and stochastic optimization. In: Proceedings of COLT. ACL, 2010
3. Eichner M, Marin-Jimenez M, Zisserman A, Ferrari V (2010) Articulated human pose estimation and search in (almost) unconstrained still images. ETH Zurich, D-ITET, BIWI, Technical Report No, 272, 2010
4. Vrigkas M, Nikou C, Kakadiaris IA (2015) A review of human activity recognition methods. Front Robot AI 2
5. Dang LM, Min K, Wang H, Jalil Piran M, Hee Lee C, Moon H (2020) Sensor-based and vision-based human activity recognition: a comprehensive survey. Pattern Recognit 107561
6. Zheng Z, An G, Wu D, Ruan Q (2020) Global and local knowledge-aware attention network for action recognition. IEEE Trans Neural Netw Learn Syst 1–14

7. Bazarevsky V, Grishchenko I, Raveendran K, Zhu T, Zhang F, Grundmann M (2020) BlazePose: on-device real-time body pose tracking, google research 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA, 17 Jun 2020
8. Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: Proceedings of international workshop on human behavior understanding. Springer, pp 29–39
9. Veeriah V, Zhuang N, Qi G-J (2015) Differential recurrent neural networks for action recognition. In: Proceedings of IEEE international conference on computer vision, 2015, pp 4041–4049
10. Shu N, Tang Q, Liu H (2014) A bio-inspired approach modeling spiking neural networks of visual cortex for human action recognition. In: Proceedings of 2014 international joint conference on neural networks (IJCNN). IEEE, 2014, pp 3450–3457
11. Weinland D, Ronfard R, Boyer E (2006) Free viewpoint action recognition using motion history volumes. Comput Vis Image Underst 104:249–257
12. Wu D, Sharma N, Blumenstein M (2017) Recent advances in video-based human action recognition using deep learning: a review. In: Proceedings of international joint conference on neural networks (IJCNN), Anchorage, AK, USA, IEEE, 2017
13. https://www.audi-electronics-venture.com/aev/web/en/driving-dataset.html
14. https://ai.googleblog.com/2019/06/announcing-youtube-8m-segments-dataset.html
15. Real E, Shlens J, Mazzocchi S, Pan X, Vanhoucke V (2017) "YouTube-BoundingBoxes: a large high-precision human-annotated data set for object detection in video. In: Proceedings of 2017 IEEE/CVF conference on computer vision and pattern recognition (CVPR 2017), arXiv:1702.00824
16. Carreira J, Noland E, Banki-Horvath A, Hillier C, Zisserman A A Short Note about Kinetics-600, arXiv:1808.01340
17. Monfort M, Andonian A, Zhou B, Ramakrishnan K, Bargal SA, Yan Y, Oliva A (2019) Moments in time dataset: one million videos for event understanding. IEEE Trans Pattern Anal Mach Intell 1–1
18. Sch¨uldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: Proceedings of 2004 international conference on pattern recognition (ICPR 2004)
19. Goyal R, Kahou SE, Michalski V, Materzynska J, Westphal S, Kim H, Haenel V, Fruend I, Yianilos P, Mueller-Freitag M et al (2017) "The" something something" video database for learning and evaluating visual common sense. In: Proceedings of 2017 IEEE/CVF conference on computer vision and pattern recognition (CVPR 2017)
20. Bhattacharyya S, Maulik U, Dutta P (2011) Multilevel image segmentation with adaptive image context based thresholding. Appl Soft Comput 11:946–962. https://doi.org/10.1016/j.asoc.2010.01.015
21. https://cocodataset.org
22. Kong Y, Fu Y (2022) Human action recognition and prediction: a survey. Int J Computer Vis 130:1-36. https://doi.org/10.1007/s11263-022-01594-9
23. Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S (2019) Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR 2019)
24. Li M, Zhou Z, Li J, Liu X (2018) Bottom-up pose estimation of multiple person with bounding box constraint. In: Proceedings of 2018 24th international conference on pattern recognition (ICPR 2018), pp 115–120
25. https://ai.facebook.com/research/publications/habitat-2.0-training-home-assistants-to-rearrange-their-habitat
26. Dang Q, Yin J, Wang B, Zheng W (2019) Deep learning based 2D human pose estimation: a survey. Tsinghua Sci Technol 24(6):663–676
27. Dimitrov D, Holst M, Knauer C, Kriegel K (2008) Experimental study of bounding box algorithms. In: Proceedings of third international conference on computer graphics theory and applications, pp 15–22

BLUEBOT—A Bluetooth Controlled Robot for Deciding Human Intervention Based on the Environment Check

Harshini Manoharan, C. R. Raghavendran, and J. Dhilipan

Abstract IoRT is the amalgamation of two cutting edge technologies Robotics and IOT that empowers intelligent robots in taking wise decisions without much human participation. In this paper, we designed and developed a Bluetooth Controlled Robot using Arduino and other components like HC-05 Bluetooth module, MQ135 Air quality gas sensor, HC-SR04 Ultrasonic sensor and water sensor and the results gathered from these sensors are used to predict the intervention of humans in a particular environment. The working process of the sensors was tested and provides us with accurate results. The navigation path of the robot is controlled using the directions (UI controls) present in the Android App developed using MIT App inventor by working with App Inventor Designer and App Inventor Blocks Editor. The results are constantly displayed to the user directly through the App that estimates the water level present in the ground, amount of toxic gas present in the atmosphere, and the distance between the robot and the obstacle. The main objective of this application is to accelerate the development of the autonomous system paradigm and the proliferation of Internet of Robotic Things by anticipating the robots in handling the situation apart from handling the well-defined task.

Keywords IoRT · HC-05 Bluetooth module · HC-SR04 Ultrasonic sensor · MQ135 Air quality gas sensor · Water sensor

1 Introduction

Industry 5.0 has created a revolution with a focus in developing the coordination between the human and the machine as human intelligence works in harmony with cognitive computing. Through Industry 5.0 humans are put back into the production

H. Manoharan (✉) · J. Dhilipan
SRM Institute of Science and Technology, Ramapuram, Chennai, India
e-mail: harshimanohar@gmail.com

C. R. Raghavendran
Easwari Engineering College, Ramapuram, Chennai, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_2

process in collaboration with the robots through which the humans can upskill and jointly reduce the time involved in the production task. By automating basic tasks that don't involve the intervention of humans, we can improve the quality and the speed of the process. The history of the Industrial revolution started off with Industry 1.0 where the production system performance was based on water and stream which was replaced with the assembly lines in Industry 2.0 and computers and electronics in Industry 3.0. With the advent of Digitalization, Industry 4.0 emerged by connective devices using different technologies including Artificial Intelligence, Data Analytics for the process of automation.

The importance of Robotics has been leveraged to a greater extent in recent years because the perception ability of the robots is in understanding its own environment through which the model is further built and upgraded. Data analytics along with sensor fusion from the IoT clearly depicts that the robots can provide a wider horizon compared to local, on-board sensing, in terms of space, time and type of information [1]. Robotics on a broader spectrum focuses on the sensors that can sense the surrounding environment and act accordingly once the data is analyzed. IOT on the other hand, focuses on utilizing sensors that could sensor and store the data either in cloud or on to the local system. IoRT all together has the dimension in combining all these components for better integration and higher capability. IoRT has already gained the attention of most of the Scientist, Industry Experts and the Academicians in developing intelligent bot capable of anticipating its own situation with the help of its perception [2].

Though the number of benefits is higher while engaging robots in the production process, still there are a lot of challenges faced in the field of Robotics that includes Reliable AI, Efficient power source, Cobots, Environment mapping, Ethics and privacy and multi-functionality. These problems can be addressed by providing thorough training to the robots so that the perception ability and the decision-making skills are increased.

2 Related Works

Robots are in increasing demand nowadays and proved to be the best solution in various sectors including Medical, Mining, Industry, Agriculture, Military operation and much more [3]. Jain et al. developed a Web Application that could control the Robot and detect for the presence of any obstacles especially while navigating along the pathway and the data is stored in cloud for later analysis [4]. RajKumar et al. in their proposed methodology made use of Wi-Fi and Internet to live stream the video of the Robot on surveillance by capturing the images through the camera placed in the Robot. The main drawback of this work is that the Robot can only capture the real time images of the environment but lacks in taking decisions and acting accordingly [5]. Rambabu et al. developed a versatile Robot capable of controlling and detecting fire automatically in disaster prone areas using Raspberry pi [6].

Surveillance robot was developed by Harshitha et al. which notifies on any trespassing and obstruction detection using Raspberry Pi 3 model. On the other hand, in the case of an authorized person the voice assistant will start talking to the robot [7]. Majority of the robots that are being developed come up with ultrasonic sensors that are highly capable in perceiving the environment and detecting the obstacle by emitting the ultrasonic sound waves [8].

Apart from surveillance, Robots were also trained and developed to carry out floor polishing tasks to ease the life of modern living using Arduino and microcontrollers. The prototype of the model includes ultrasonic sensor for obstruction detection, fans, motors, discs and LED controlled through the mobile Bluetooth [9]. With the advantage of programming and reprogramming, the Arduino board has given the flexibility to program the hardware components along with the software's without much difficulty [10]. Temperature and humidity sensors are embedded into the robots to monitor the changes in the atmospheric environment and the data is transmitted through a wireless transceiver module [11].

The role of Robots in the field of waste management has extended up to a great extent where intelligent bots were built that could alert the Municipal Web server when the level of garbage becomes full. This way we can pave the way for an effective waste management system and provide a room for smart cities [12]. Most existing works focused on Bot creation for surveillance, fire extinguishing, floor cleaning and waste management. Our work was developed with a strong intention to help the Military officials and other people in a view to save the life of many. This Bot is sent for the environment check to verify if the environment is safe for human intervention or not. The data is collected and stored securely based on which the decisions are taken accordingly.

3 Methodology

The proposed prototype is designed with the objective to develop an intelligent device that can sense the surrounding environment and activities through the embedded sensors to take necessary actions without much human intervention. The robots are well trained to take up the decision on its own. Figure 1 demonstrates the working of the Bluebot, the initial step is to connect the Bluetooth module of the Robot to the mobile Bluetooth, through which the communication between the Robot and the Application can be established continuously. On successful connection, the Bluebot starts sensing its environment through the embedded sensors that can detect the harmful gas present in the atmosphere, and water level present at the ground. The Bluebot is designed to navigate through the environment which is controlled by the UI present in the Android Application. During the process of navigation, the ultrasonic sensors detect for the presence of any obstacle and automatically stop from further navigation if the robot encounters any obstacle.

Figure 2 shows how the hardware circuits are connected for establishing continuous communication. Two motors M1 and M2 as shown in Fig. 3 are connected

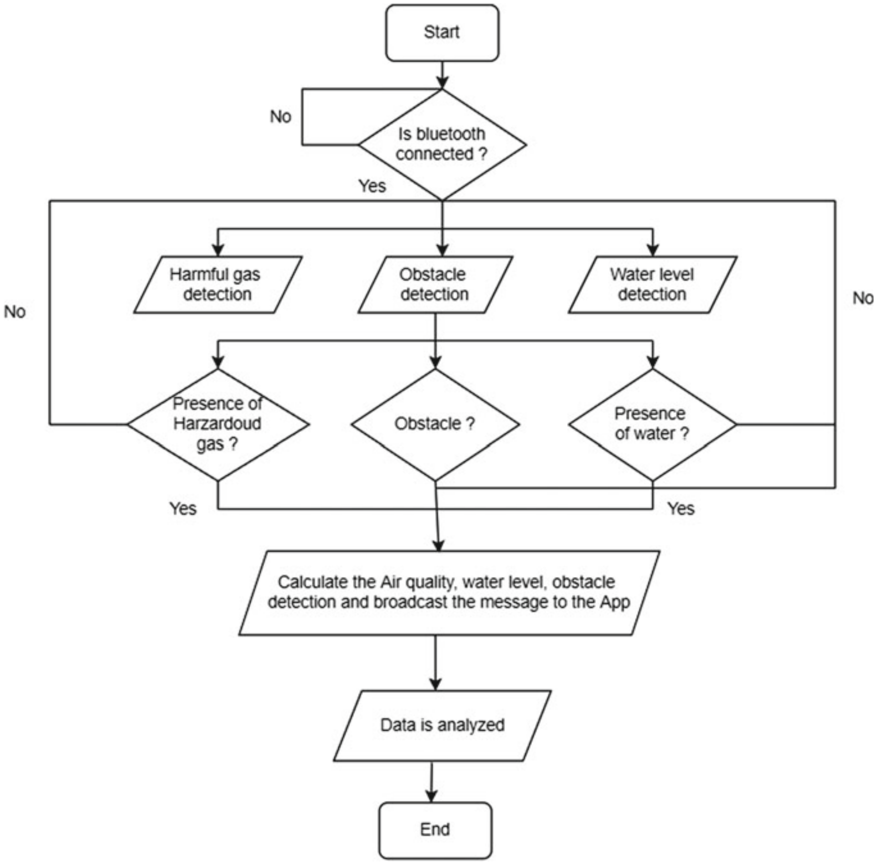


Fig. 1 Flow chart of Bluebot

to the motor driver for the easy movement of the Robot. Power supply is provided through the battery that is rechargeable.

3.1 Development of the App

An Android application that can control the movements of the system was developed using the MIT App Inventor. The MIT App Inventor was developed by Google and maintained by Massachusetts Institute of Technology is free and open source. The apps are built by working with the App Inventor Design, where the components of the App are designed and the App Inventor Block Editor, where we assemble programs visually through drag and drops. Blocks also referred to as virtual elements are linked with one another using the logical statements. The data from the sensors are received

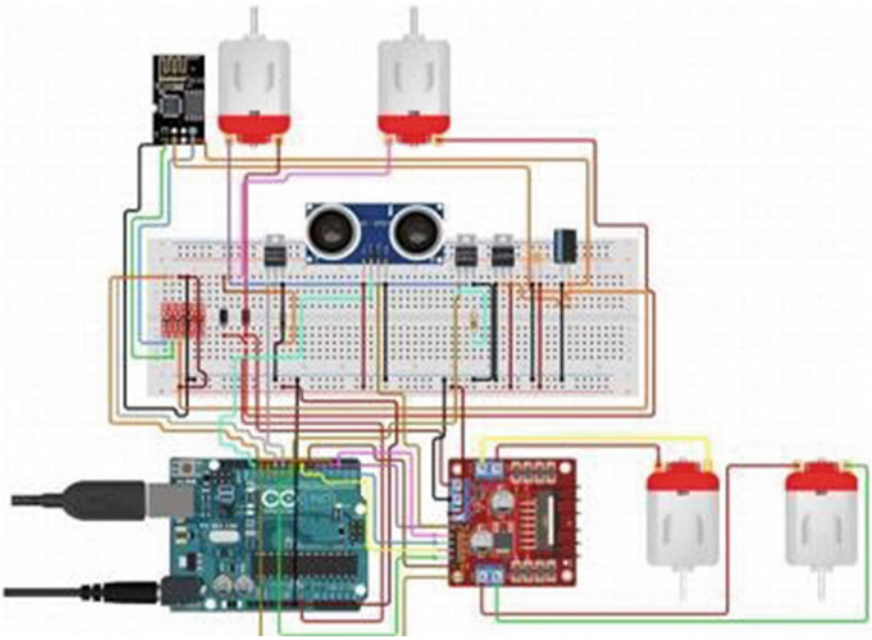


Fig. 2 Hardware circuit design

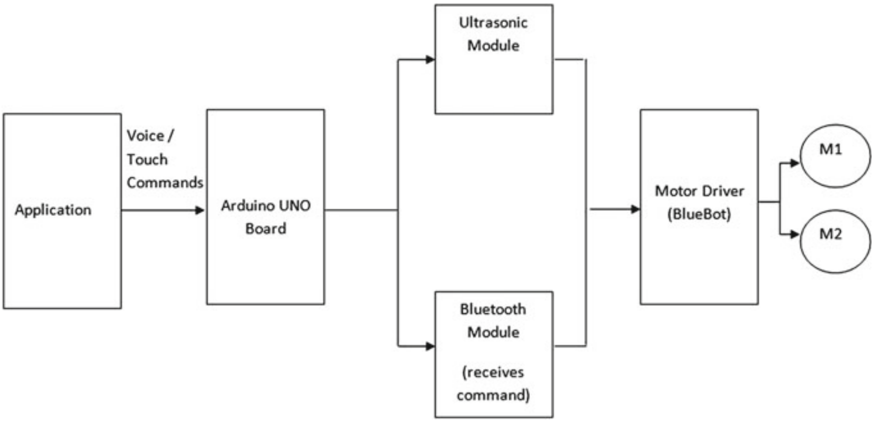


Fig. 3 Block diagram of Bluebot

and displayed through this application through the Bluetooth module embedded in the Bluebot.

3.2 Arduino UNO

Arduino Uno, microcontroller board based on the ATmega328P, is designed with 14 digital input and output pins, 16 MHz quartz crystal, 6 analog inputs, USB connection, a power jack, an ICSP header and a reset button [13]. The board is composed of everything needed to support the microcontroller and gets connected to the computer with a USB cable or powered with an AC-to-DC adapter or battery to keep running.

3.3 Bluetooth Module

The Bluetooth module HC-05 module is intended to work as Master/Slave module which is configured only through AT COMMANDS. By default, the module is set to SLAVE. AT Commands are utilized to modify the basic configuration of the module. HC-05 Module is composed of 5 or 6 pins. The firmware for HC04 is LINVOR and for the HC05 it is HC05 itself. On scanning the Bluetooth devices from the App the name becomes visible on the Android phone.

3.4 Ultrasonic Module

HCSR04 Ultrasonic Sensor is used to detect the presence of obstacles by emitting high frequency sound waves which are too loud for humans to hear. These sound waves hit the obstacle and get reflected, which calculates the distance based on the time required. The calculated distance is then displayed in the application of the user. When the distance between the obstacle and the system becomes less than 20 cm, the system stops moving in that direction.

3.5 Gas Sensor

Gas Sensor is embedded into the robot which has the potential to detect toxic gases including ammonia, nitrogen oxide, CO_2, oxygen, alcohols, aromatic compounds, sulfide and smoke present in the surrounding environment. The conductivity of the gas sensor leverages as the concentration of the polluting gases increases. A 5 V power supply is provided to the sensor.

3.6 Water Level Sensor

A level sensing device is used to measure the level of flow of substance including liquids, slurries, and granular materials. A 5 V power supply is provided to the sensor. The sensor is calibrated to various types of water to get out the accurate readings.

3.7 Motor Driver

Motor M1 and M2 are connected to the control circuits through the motor driver. L293D is a motor driver IC that allows DC motors to drive in either direction. All the components of the robot are enclosed within the chassis.

4 Result

4.1 Test Report

A very rudimentary approach with testing of bots is that the test condition under all combinations of inputs and preconditions (initial state) is not expedient, even with a minor product. The number of obstructions in the developed product is substantial with defects occurring infrequently are arduous to figure out in testing. To meet the non-functional requirements of the developed product including quality, usability, scalability, performance, compatibility, reliability are highly subjective, and sporadically sufficient value to one person may not be tolerable to other person. We plan to test our robot by profuse methods of testing. We aim at testing our robot and run it under maximum testing combinations possible. By doing this we can examine the functionality and the durability of our robot. Testing of the Robot is categorized into three phases, being unit testing, integration testing and validation testing. Integration testing is the most detailed and longest process of testing as it consists of the top-down approach, bottom-up approach, umbrella approach, black box testing and white box testing.

4.1.1 Test Description TC01

The developed Android Application is installed on the Android phone that acts as a controller for the system. The Bluetooth Communication between the Arduino and Android Phone is enabled by the HC-05 Bluetooth Module. The user should be able to view all Bluetooth devices by clicking the button (Tables 1 and 2).

Table 1 Test case 1 information

Test case ID	TC01
Test case name	Bluetooth connection check
Test case objective	Connect to the Bluetooth module present in the system

Table 2 Test case TC01 report

Step	Test steps	Test data	Expected result	Actual result	Result
1	Connect	Button click	List of available Bluetooth devices	List of all paired and available Bluetooth devices are displayed	PASS
2	Disconnect	Button click	Bluetooth is disconnected	The connected Bluetooth module id disconnected, and the system stops responding	PASS

4.1.2 Test Description TC02

The Motor Driver Module L298N enables the motors of the robotic car to drive through the current flow. On key press, the data is automatically sent to the Bluetooth module by enabling the Bluetooth connection. Each switch case is mapped with instructions to the Motor Driver Input Pins in the Arduino software, which is made to receive data from the Bluetooth Module and perform a straightforward switch case operation. For instance, on pressing UP Arrow in the App, ‘F’ is transmitted. Arduino causes the wheels to drive ahead by setting IN1 and IN3 to HIGH and IN2 and IN4 to LOW. Corresponding to this, other keys match the correct IN1–IN4 pin configurations (Tables 3 and 4).

Table 3 Test case 2 information

Test case ID	TC02
Test case name	System controller check
Test case objective	Controlling the system by connecting the Bluetooth placed in the system with the Bluetooth of the Android phone through the App

Table 4 Test case TC02 report

Step	Test steps	Test data	Expected result	Actual result	Result
1	Forward	F	Move forward	System moves forward	PASS
2	Backward	B	Move backward	System moves backward	PASS
3	Left	L	Move left	System moves left	PASS
4	Right	R	Move right	System moves right	PASS
5	Stop	S	Stop moving	System stops moving	PASS

Table 5 Test case 3 information

Test case ID	TC03
Test case name	Data retrieval check
Test case objective	Retrieving the data from various sensors and displaying it on the App

Table 6 Test case TC03

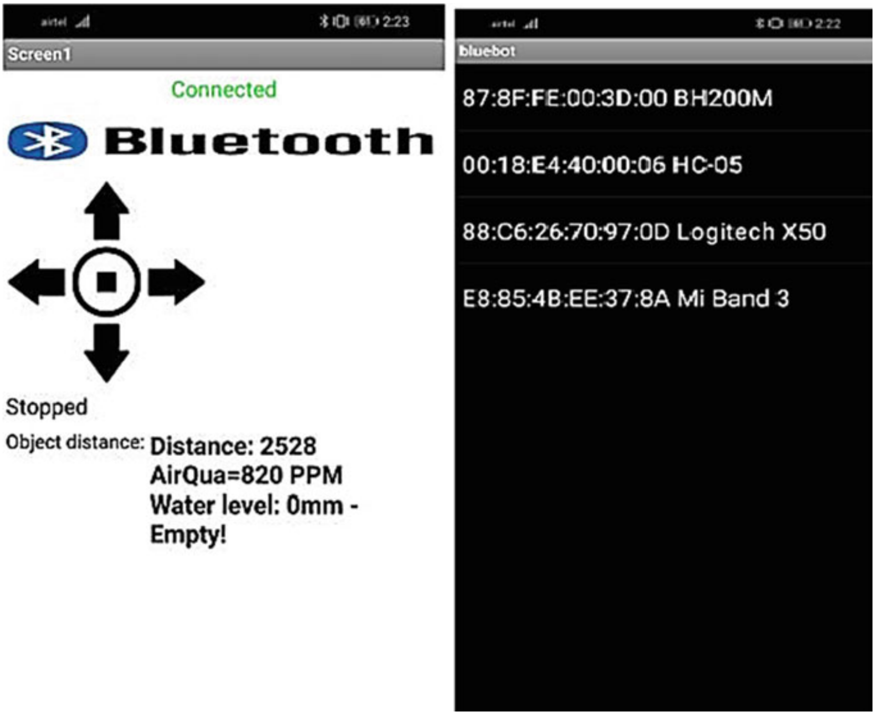
Step	Test steps	Test data	Expected result	Actual result	Result
1	Obstacle detection	Conditions around	Distance between the obstacle and the system should be detected and displayed	Distance between the obstacle and the system is detected and displayed in the Android App	PASS
2	Water level detection	Presence of water in the environment	Water level should be detected, and the readings are displayed in the Android App	Water level is detected ad the readings are displayed in the Android App	PASS
3	Air quality detection	Presence of harmful/harmless gas in the environment	Quality of the air should be detected and displayed in the Android App	Quality of the air is detected and displayed in the Android App	PASS

4.1.3 Test Description TC03

The data received from the sensors are displayed in the serial monitor. To make it available in the Android App, data in bytes is received through the Bluetooth Module that is connected. The data is then displayed in the empty fields that are designed in the App (Tables 5 and 6).

5 Conclusion and Future Scope

IoRT based BlueBot is basically a robot built using the Bluetooth module and controlled using an Android App. During this work a successful working model of the robot was constructed as shown in Fig. 4, which has three sensors embedded in it namely the water sensor, capable of sensing the water level when the robot is sent into the underground tunnels. Secondly the gas sensor measures the amount of toxic gas present in the atmosphere. Finally, the presence of the ultrasonic sensors is utilized for predicting the distance between the robot and the obstacle so that the user can make necessary decisions on navigation through the path. The communication between the robot and the Android App is established by connecting the Bluetooth module with the mobile's Bluetooth. Furthermore, the robot runs on a battery by supplying the power to the motors. The Bluetooth module used in this work has



Connected

Stopped

Object distance: Distance: 2528
AirQua=820 PPM
Water level: 0mm -
Empty!

87:8F:FE:00:3D:00 BH200M

00:18:E4:40:00:06 HC-05

88:C6:26:70:97:0D Logitech X50

E8:85:4B:EE:37:8A Mi Band 3

Fig. 4 Android App used to control the system and display the data received from the sensors

a very limited range covering up to 10 m that can be modified by using the LoRa module as a future scope. Additionally a servo motor can be added which redirects the movement of the robot as soon as it meets the obstacle. A 360° camera is embedded to the module that can help the user in visualizing the movements of the robot.

References

1. Simoens P, Dragone M, Saffiotti A (2018) The Internet of Robotic Things: a review of concept, added value and applications. Int J Adv Robot Syst 15:1–11. https://doi.org/10.1177/1729881418759424
2. Romeo L, Petitti A, Marani R, Milella A Internet of Robotic Things in smart domains: applications and challenges. Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA), National Research Council (CNR), Bari, Italy. https://doi.org/10.3390/s20123355
3. Trisha, Deepak Kumar S (2020) Design and development of IoT based robot. In: 2020 international conference for emerging technology (INCET), Belgaum, India, 5–7 June 2020. https://doi.org/10.1109/INCET49848.2020.9154175

4. Jain RK, Saikia BJ, Rai NP, Ray PP (2020) Development of web-based application for mobile robot using IOT platform. In: 2020 11th international conference on computing, communication and networking technologies (ICCCNT). IEEE. https://doi.org/10.1109/ICCCNT49239.2020.9225467
5. RajKumar K, Saravana Kumar C, Yuvashree C, Murugan S (2019) Portable surveillance robot using IOT. Int Res J Eng Technol (IRJET) 6(3). e-ISSN: 2395-0056
6. Rambabu K, Siriki S, Chupernechitha D, Pooja Ch (2018) Monitoring and controlling of fire fighting robot using IOT. Int J Eng Technol Sci Res 5(3). ISSN 2394-3386
7. Harshitha R, Hameem M, Hussain S (2018) Surveillance robot using Raspberry Pi and IoT. In: 2018 international conference on design innovations for 3Cs compute communicate control (ICDI3C). IEEE. https://doi.org/10.1109/ICDI3C.2018.00018
8. Zhmud VA, Kondratiev NO, Kuznetsov KA, Trubin VG, Dimitrov LV Application of ultrasonic sensor for measuring distances in robotics. J Phys: Conf Ser 1015(3)
9. Goon LH, Md Isa ANI, Choong CH, Othman WAFW (2019) Development of simple automatic floor polisher robot using Arduino. Int J Eng, Creat Innov 1(1)
10. Badamasi YA (2014) The working principle of an Arduino. In: 2014 11th international conference on electronics, computer and computation (ICECCO). IEEE. https://doi.org/10.1109/ICECCO.2014.6997578
11. Wang Y, Chi Z (2016) System of wireless temperature and humidity monitoring based on Arduino Uno platform. In: Sixth international conference on instrumentation & measurement, computer, communication and control (IMCCC). IEEE. https://doi.org/10.1109/IMCCC.2016.89
12. Sathish Kumar N, Vuayalakshmi B, Jenifer Prarthana R, Shankar A (2016) IOT based smart garbage alert system using Arduino UNO. In: 2016 IEEE Region 10 conference (TENCON). IEEE. https://doi.org/10.1109/TENCON.2016.7848162
13. Louis L (2016) Working principle of Arduino and using it as a tool for study and research. Int J Control, Autom, Commun Syst (IJCACS) 1(2). https://doi.org/10.5121/ijcacs.2016.1203

Nirbhaya Naari: An Artificial Intelligence Tool for Detection of Crime Against Women

Sakshee Sawant, Divya Raisinghani, Srishti Vazirani, Khushi Zawar, and Nupur Giri

Abstract Dowry abuse, rape, domestic violence, forced marriage, witchcraft-related abuse, honor killings are just a few of the myriad atrocities women encounter and fight against worldwide. The psychological impacts of abuse on the victim can lead to depression, PTSD, eating disorders, withdrawal from the outside world and society, and low self-esteem to name a few. The physical implications could result in an inability to get to work, wage loss, dearth of involvement in routine activities, not being able to take care of themselves and their families. Our initiative is dedicated to curbing violence against women by providing a forum for women to speak about violence as well as passing a signal about it through a dedicated hand gesture. Our designed solution has three modules namely: Violence/Crime Scene Detection against women using audio and video, help hand signal detection, and multi-label story classification. Our approach uses Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) for video classification along with Support Vector Machine (SVM), and Random forest for audio classification.

Keywords Violence detection · Abuse · Harassment · Hand gestures · Crime · Residual networks (ResNets) · Convolutional neural network (CNN) · Deep learning · Word embeddings

S. Sawant (✉) · D. Raisinghani · S. Vazirani · K. Zawar · N. Giri
Department of Computer Engineering, Vivekanand Education Society's Institute of Technology, Mumbai 400074, India
e-mail: 2018.sakshee.sawant@ves.ac.in

D. Raisinghani
e-mail: 2018.divya.raisinghani@ves.ac.in

S. Vazirani
e-mail: 2018.srishti.vazirani@ves.ac.in

K. Zawar
e-mail: 2018.khushi.zawar@ves.ac.in

N. Giri
e-mail: nupur.giri@ves.ac.in

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*, Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_3

1 Introduction

The United Nations defines violence against women as "any form of gender-based violence that causes or is likely to cause physical, sexual, or mental harm or suffering to women, including violence, compulsion, or arbitrary deprivation of liberty, regardless of it being conducted in public or in private [1]". WHO, on behalf of the "United Nations Interagency Working Group on Violence Against Women," studied a data from 2000 to 2018 across 161 nations and territories concluding that on an average one in three i.e. approximately 30% women have been subjected to violence by a partner [2]. Of the several crimes happening against women globally, domestic violence tops the chart. From the outset, the Indian Constitution granted equal rights to men and women to abolish gender disparity. Unfortunately, most women in this country are unaware of their rights due to illiteracy and prejudiced mindsets [3]. This lack of awareness about women's rights keeps them on the receiving end resulting in a lowering of their self-esteem, decreasing their productivity and performance eventually [4]. These seemingly minute things can result in cascading effects such as homicide, suicide, death, unintended pregnancies, induced abortions, gynecological issues, STDs, eating and anxiety disorders, and many more [5].

In an effort to address this pressing issue, this paper attempts a model "Nirbhaya Naari", a model that can detect potential violent actions against women and ensure women's safety. Identification of violence/crime situations against women, audio–video conversion, detection of verbal harassment or verbal help, story summarization, and detection of help hand signals are the four key models that we have worked upon. The Audio Summarization model takes into account the user's version of the abuse and determines the type of violence the user experienced. Depending on the nature of the abuse, the appropriate help is extended. The video's Violence/Crime Scene Detection Module recognizes any conflicts or physical assaults directed toward women. Audio detection of Verbal Harassment or Verbal Help detects the presence of help screams in video footage. The Help Hand Signal Detection module looks for help hand signals in video frames. The audio, video, and story proofs of violence are stored in Aws Bucket and the aws S3 url is recorded in php database.

The novel aspect of our proposal is that it adopts a heuristic approach to the victim in order to identify any form of injustice action against women itself rather than just the presence or absence of violence by considering more factors like screams, violent actions, and more input types like audio, video, text format and help hand signal. Detection of female screams specifically, adds to the confidence score and thus, accuracy and chances of detecting violent activity increase. The portal strives to assist women who are victims of abuse or violence by giving them a safe place to seek aid and ask questions. A portal that could be used by the victim or the witness. All of this would be accomplished with the help of a collection of algorithms, with the sole goal: WE WANT WOMEN TO FEEL SAFE (Fig. 1)!

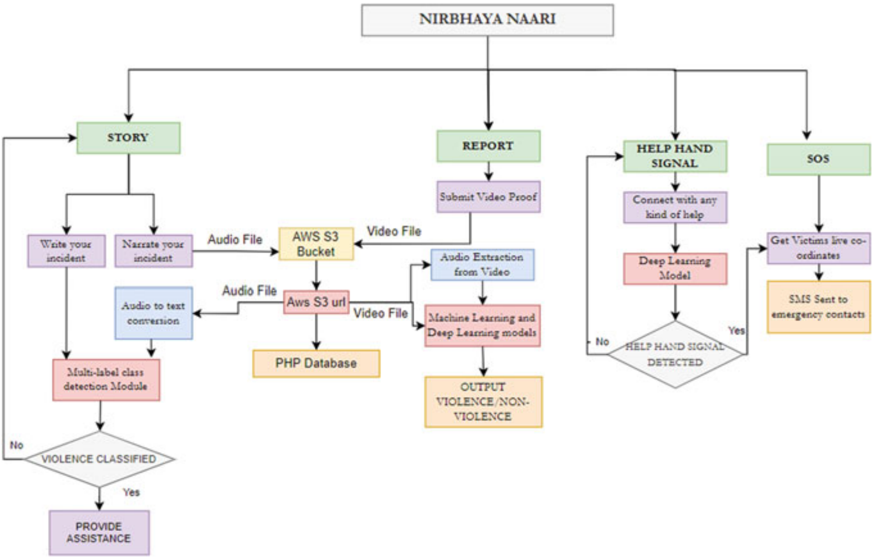


Fig. 1 Functioning of “Nirbhaya Naari”

2 Related Work

The importance of security construction has risen significantly since the concept of a “safe city,” and video monitoring technology has been continually explored and implemented. Video surveillance systems are required to grow more intelligent since the functional needs of actual applications become more diversified. Reference [6] investigates a specialized intervention strategy for IPV victims (aged 18 to 65 years) in the emergency room that allows treatment from professionals. The aim was to figure out if there was a link between increasing screening rates and greater detection of violence. Reference [7] studies cases of domestic violence among admissions to a large Italian hospital’s emergency room in 2020, including during the whole “Lockdown”. It also focuses on documenting the short and long-term health effects of violence, as well as evaluating the WHO screening as a tool for finding instances that might otherwise go unnoticed. Reference [8] focuses on the relationship between adult patients’ socioeconomic factors and domestic violence screening and further help-seeking behavior if they are assaulted. The findings showed that basic and primary care can play a key role in recognizing domestic abuse by using the “WAST (Women Abuse Screening Tool)” screening instrument, or a suitable adaptation of it. Reference [9] is about eve-teasing in rural areas, particularly among female adolescents, and suggests a method for determining its prevalence. Direct observation of questionnaires, group discussions, and semi-structured interviews was used as part of a mixed technique study. Reference [10] proposes an architecture that extracts features from video frames using a pre-trained “ResNet-50 model”, which is then fed into a “ConvLSTM block”. To eliminate occlusions and inconsistencies, a short-term difference in video frames to provide additional robustness is used. “Convolutional neural networks allow us to extract more concentrated Spatio-temporal data from frames, which helps LSTMs cope with the sequential nature of films”. The model takes raw movies as input, translates them into frames, and then generates a binary label of violence or non-violence. To remove extraneous details, we used cropping, dark-edge removal, and other data augmentation techniques to pre-process the video frames. Reference [11] claims that the technique extracts the video’s spatiotemporal aspects using artificial features and depth features, which are then integrated with the trajectory features using a convolutional neural network. Face images in surveillance video cannot be successfully recognized due to low resolution, so the multi-foot input CNN model and the SPP-based CNN model were created to address this problem. The accuracy of the brute force identification approach suggested in this study was tested on the ‘Crow and Hockey datasets’, and it was shown to be as high as 92 percent and 97.6 percent, respectively. The violence detection method suggested in this research improves the accuracy of video violence detection, according to experimental data.

Reference [12] introduces a new audio classification technique. For audio classification, the initial step is to propose a frame-based multiclass support vector machine (SVM). This novel audio feature is combined with ‘mel-frequency cepstral coefficients (MFCCs)’ and five perceptual features to produce an audio feature set. Reference [13]. The main goal of this work is to use hand photos acquired with a webcam to detect American sign characters. The Massey dataset had a success rate of 99.39 percent, the ASL Alphabet dataset had a success rate of 87.60 percent, and the Finger Spelling A dataset had a success rate of 98.45 percent as in Reference [14]. This study aims to: (1) develop a new “gold standard” dataset from social media with multi-class annotation; (2) execute extensive experiments with several deep learning architectures; (3) train domain-specific embeddings for performance improvement and knowledge discovery; and (4) generate visuals to facilitate model analysis and interpretation. Empirical research using a ground truth dataset has shown that class prediction can be as accurate as 92 percent. The study validates the application of cutting-edge technology to a real-world situation, and it benefits in DVCS organizations, healthcare providers, and, most crucially, victims.

3 Dataset Source and Data-Preprocessing

We collected the suitable, curated, and most delinquent data from Facebook as the primary social media channel because of its widespread popularity and considerable interaction of sharers and supporters on Facebook groups. The entries were gathered from a variety of pages that examine various aspects of violence. The extraction technique employed the Facebook Graph API, and the search phrases were ’domestic violence and ‘sexual violence [14].’ The manual classification of data extracted was done in order to create a benchmark. Because human annotation is a time-consuming process, we randomly chose 2,000 postings. We kept cases including hyperlinks or

Table 1 Example of posts and their respective category

Post ID	Posts	Label
1	"My brother had molested me once when I was ten years old. My subconscious hid this trauma to allow me to live under the same roof with someone who had violated my body and my trust."	Sexual violence
2	"I found myself in the position I never expected to be in, echoing the words of countless women undone by the violence of the men in their lives: 'But I still love him." #DomesticAbuse #WhyIStayed	Domestic violence
3	"I had no money a 3-month-old baby and he controlled money I was scared my baby would go hungry." #whyistayed #domesticabuse #coercivecontrol	Psychological violence
4	A 13-year-old girl was raped and killed, in Gurgaon (Gurugram district, Haryana), by a relative of their landlord belonging to a general caste. 1 dead	Fatalities

graphics out of the analysis. The total number of postings in the final benchmark corpus was 1064, divided into four categories: sexual violence, domestic violence, psychological violence, and fatalities. Given that no earlier study on multi-label violence identification has been done, the size of the acquired dataset is considered moderate.

- S1 post as Sexual violence: Women sharing personal experiences to educate other women
- S2 post as Domestic violence: Women sharing if they are facing/overcoming domestic abuse
- S3 post as psychological violence: Financial crisis and verbal abuse taking a toll on mental health
- S4 post as Fatalities: Women sharing experiences where they have lost a loved one to abuse (Table 1).

The use of word embeddings as features extraction is a key aspect of multi-label identification tasks. Word embeddings are a more expressive way of representing text data since they capture the relationships between the terms (Fig. 2).

Along with this data, we also collected video data to increase the accuracy of our model. The two pertinent factors to scrutinize here are the sight of violent actions and screams. The proposed model is trained and evaluated on a violent crowd data-set, fight dataset, hockey dataset containing fight scenes, and audio data ie. screams scraped from the internet [15, 16]. For crime detection, via videos, a total of 1000 videos of 720 × 576 pixels were hand-labeled as "violent" or "non-violent". The complete audio data set was separated into two classes for the detection of violence. A positive class contains roughly 2000 human screams that are further classified into male and female screams and a negative class has around 3000 negative sounds that are not considered screams. For further analysis, we also incorporated a gesture detection dataset to spread alerts.

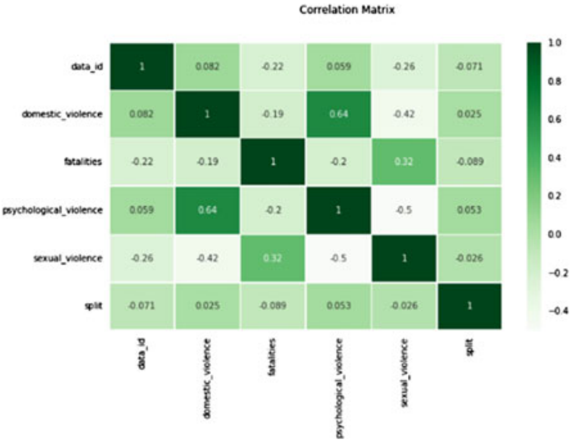


Fig. 2 Correlation matrix of the dataset features for multi-label classification

For silently seeking help, women can raise their hands. For this, two datasets have been utilized. The ‘Canadian Women’s Foundation’ created the Signal for Help, which was launched on April 14, 2020. After the Women’s Funding Network (WFN) embraced it, it quickly expanded around the world. The signal is made by displaying the ASL characters A and B. To begin, the Kaggle ASL alphabet dataset was used for indication detection to assess the performance of more intricate data. The NUS hand posture dataset with the cluttered background was used to compare the results to earlier research and determine which had the highest recognition rate [17]. For a deeper comprehension of the considered datasets, the figure exhibits examples from them, one with a plain background and the other with a cluttered background (Fig. 3).

The ASL alphabet dataset consists of 780,00 images, containing 300 samples per class. We used 600 images for it because we needed to train the model only on two specific signs that denote help. And for the NUS hand posture dataset with the cluttered background, we used 400 images. Our approach, which is further explained in detail in the methodology, eliminates the time-consuming task of identifying prospective feature descriptors capable of determining different gesture types.

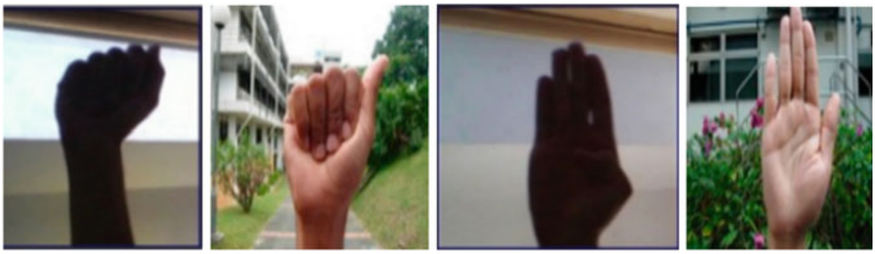


Fig. 3 Sample images for the ASL alphabet dataset and NUS hand posture dataset respectively

4 Methodology

The model's functionality is broken down into three stages. The first phase identifies violent behaviors, the second detects cries/screams, particularly female screams, and the third phase recognizes the existence of a help hand gesture. When analyzing a video sequence, the output of the three stages is taken into account. Phase 1 and phase 2 outputs are assessed simultaneously, with phase 1 output getting higher weight. Phase 3 is viewed independently from phases 2 and 1. If violence is detected in Phase 1 or Phase 2 the submitted audio/video file is stored in Aws S3 bucket. The Aws S3 url is then stored in the php database. If a help hand signal is detected live location coordinates are obtained of the victim and a SOS text message is sent to emergency contacts. The former depicts the presence of violence, whilst the latter depicts a request for assistance (Fig. 4).

4.1 Phase 1 and 2: Violence/Crime Scene Detection Using Video/Audio

We ascertain two diverse but allied tasks with a video sequence Sq comprising frames {frame1, frame2,frame3,...}. The video Sq is presumed to be segmented temporally, encasing T frames delineating either violent or non-violent behavior. The intent is to assort S accordingly. The frames are read in four-dimensional tensors (frame, H, W, RGB). A base model built employing pre-trained CNNs (vgg-19) is applied, succeeded by LSTM cells and dense layers. We use 700 videos for the training set and 300 videos for the test set in the base model. Transfer Learning is employed to minimize computation power consumption and improve accuracy.

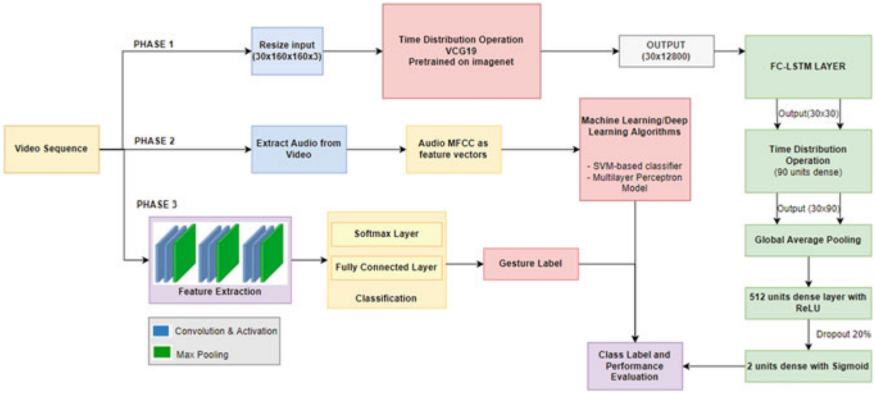
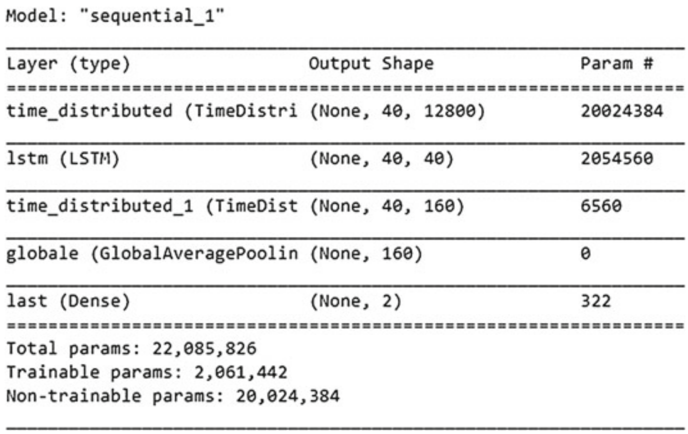

Video Sequence
PHASE 1
Resize input (30x160x160x3)
Time Distribution Operation VCG19 Pretrained on imagenet
OUTPUT (30x12800)
FC-LSTM LAYER
Output(30x30)
Time Distribution Operation (90 units dense)
Output (30x90)
Global Average Pooling
512 units dense layer with ReLU
Dropout 20%
2 units dense with Sigmoid
PHASE 2
Extract Audio from Video
Audio MFCC as feature vectors
Machine Learning/Deep Learning Algorithms
- SVM-based classifier
- Multilayer Perceptron Model
PHASE 3
Feature Extraction
Softmax Layer
Fully Connected Layer
Classification
Gesture Label
Class Label and Performance Evaluation
Convolution & Activation
Max Pooling

Fig. 4 Architecture diagram



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
time_distributed (TimeDistri (None, 40, 12800)         20024384
_________________________________________________________________
lstm (LSTM)                  (None, 40, 40)            2054560
_________________________________________________________________
time_distributed_1 (TimeDist (None, 40, 160)           6560
_________________________________________________________________
globale (GlobalAveragePoolin (None, 160)               0
_________________________________________________________________
last (Dense)                 (None, 2)                 322
=================================================================
Total params: 22,085,826
Trainable params: 2,061,442
Non-trainable params: 20,024,384
_________________________________________________________________
```

Fig. 5 Model architecture

To detect women's screams we primarily extract audio from video frames and transform it to Mel Frequency Cepstral Coefficient, a feature popularly used in automatic speech recognition, using the Librosa package. We trained the SVM (Support Vector Machine) model to apprehend human screams/shouts using the MFCCs of the audio retrieved from supplied input. The MPN (Multilayer Perceptrons Model) was then trained to detect female screams particularly.

The following are the steps we took in general:

1. As input, video is used.
2. Extract Audio from Video.
3. In the preprocessing phase, resize the video and get the audio MFCC coefficients
4. The preprocessed data is then fed into SVM, MPN, and CNN models.
5. Ascertain whether violence and women's screams were detected (Fig. 5).

4.2 Phase 3: Help Hand Signal Detection

Further, the model has an input layer and convolution layers that are used to apply a set of convolution filters to an image. The layer executes a sequence of mathematical operations for each subregion to create a single value in the feature map, pooling layers downsample the image data retrieved by the convolutional layers to reduce the dimensionality of the feature map to reduce the processing time, while convolutional layers apply a ReLU activation function to the output to inject non-linearities into the model. We employed max pooling, which takes feature map subregions (100 × 100-pixel tiles), preserves their maximum value, and discards the rest, one softmax output layer, and a final interconnected output layer [17]. The input image of hand postures is received by the input layer, which then sends them on to the subsequent layers for extracting features and classification. The proposed design has three convolutional layers: eight 19 × 19 filters in the first layer, sixteen 17 × 17 filters in the second layer, and thirty-two 15 × 15 filters in the third. Now we fed the output of the convolution process into a succession of ReLu activation neurons [18]. Zero takes the place of negative values in the pooling layer by using the non-linearity & non-saturating function defined in the equation:

$$y = \max(0,\ x) \tag{1}$$

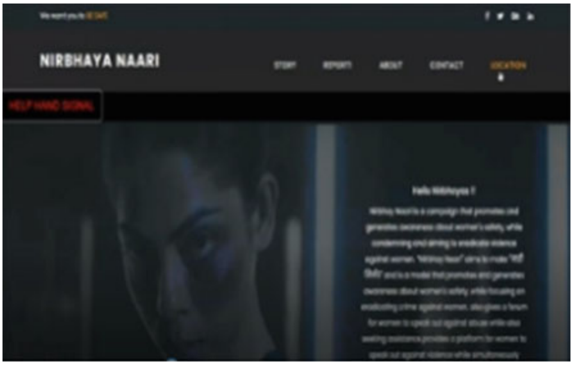
The classification layer receives the most discriminatory feature values recovered by the multiple-layered structure. The softmax layer and the output layer have the same number of neurons, the output layer's number of neurons is determined by the number of classes in the dataset, and also it uses a multiclass sigmoid function to restate the feature values into the range 0–1. We get a feature vector from this layer. Based on this, the final fully connected layer classifies the input frames into the appropriate gesture and the system will spread an alert if a help hand gesture is detected.

4.3 Multi-label Classification of Stories

Now, if the environment is not under scrutiny and some user wants to share their story, the user can login to our portal and can write or narrate her story. After processing the information, the system will connect the user to the nearest help centers. There are two techniques to input embeddings to neural networks in multi-label classification, which are employed here to determine the sort of assault a woman has encountered—by training, by using domain-specific word embeddings (Keras), and by making use of pre-trained embeddings (e.g. Word2vec, FastText, and Glove). The models that we delved into using domain-specific embeddings are NN, CNN, RNN, and LSTM. And the models we investigated using pre-trained embeddings (with Word2Vec, FastText, and Glove) are CNN, RNN, and LSTM.

In this, we used the training dataset as our (train and validation data), while the test dataset was our test data. For the train data, we used cross-validation to split the train data into random train and validation subsets. Our model is then iteratively trained and validated on these different sets.

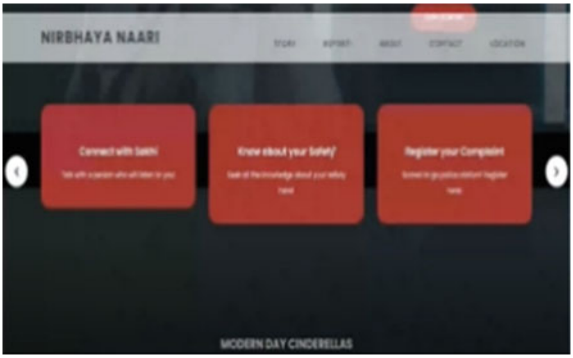
The training dataset is split into train and validation. We had a training split, 0.8 of for train dataset samples, while the validation split, was 0.2 for Train dataset samples. We calculated the precision, a multi-label classification parameter that indicates how many relevant things were chosen, as well as the recall, a multi-label classification metric that assesses how many significant items were chosen [19]. Eventually, the F-score viz weighted harmonic mean of recall and precision, was calculated. This is critical for multi-label classification, which groups input samples into label sets. A model would attain a perfect score by merely allocating every class to every input if it just considered the accuracy (precision). To circumvent this, a measure should also penalize erroneous class assignments (recall). This is determined by the F-beta



Hello Nirbhayaa !

Nirbhay Naari is a campaign that promotes and generates awareness about women's safety, while condemning and aiming to eradicate violence against women. "Nirbhay Naari" aims to make "All Safe" and is a model that promotes and generates awareness about women's safety while focusing on eradicating crime against women. Also gives a forum for women to speak out against abuse while also seeking assistance provides a platform for women to speak out against violence while simultaneously

Fig. 6 Homepage of portal "Nirbhaya Naari"



Talk with a person who will listen to you

Gain all the knowledge about your safety here

Unable to go police station? Register here

Fig. 7 Additional functionalities of our portal

score [20]. This is identical to an F-measure when beta = 1. We employed domain-specific embeddings along with pre-trained embeddings. The analysis revealed that domain-specific embeddings outperform pre-trained embeddings.

5 Implementation Screenshots

See Figs. 6, 7, 8, 9, 10, 11, 12 and 13.

6 Result Analysis

Hand segmentation, which is actually a tough operation in photographs with backgrounds containing different items, is not required with the suggested model. Even though there are different segmentation approaches available based on skin color, hand shape, and various other factors, they all fail to produce accurate results when

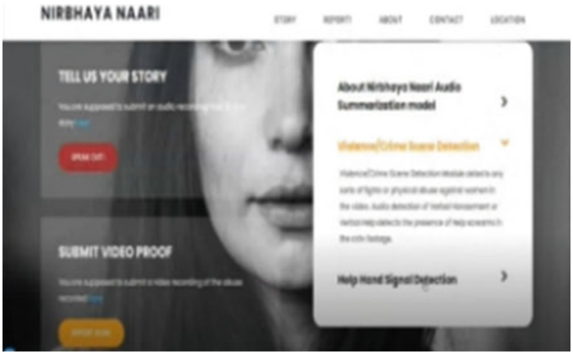

NIRBHAYA NAARI
STORY
REPORT
ABOUT
CONTACT
LOCATION
TELL US YOUR STORY
SPEAK OUT!
SUBMIT VIDEO PROOF
About Nirbhaya Naari Audio Summarization model
Violence/Crime Scene Detection
Help Hand Signal Detection

Fig. 8 Section for victim to provide audio/video proof of her violence

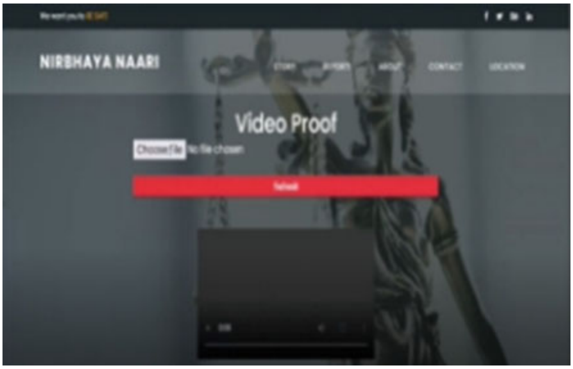
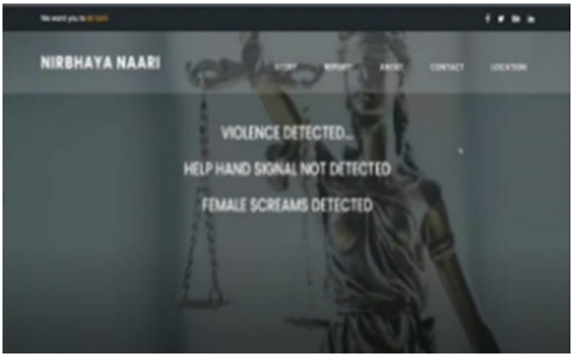


Fig. 9 Section for victim to upload a video proof of her abuse



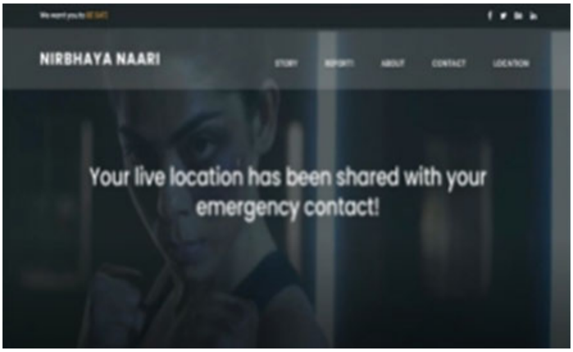
VIOLENCE DETECTED..

HELP HAND SIGNAL NOT DETECTED

FEMALE SCREAMS DETECTED

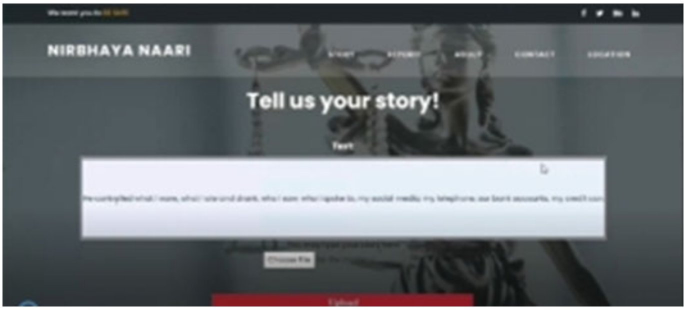
Fig. 10 Portal detecting the presence of violence

applied to photos with other background items or moving items. Our suggested method also removes the time-consuming task of identifying prospective feature descriptors capable of distinguishing different gesture types. We tried and tested different models which are: CNN with the highest accuracy of 96.42%, Inception V3 with an accuracy of 90%, DNN with Squeezenet with an accuracy of 83.28%,



Your live location has been shared with your emergency contact!

Fig. 11 Sharing live coordinates of victim with her emergency contact when system detects help hand



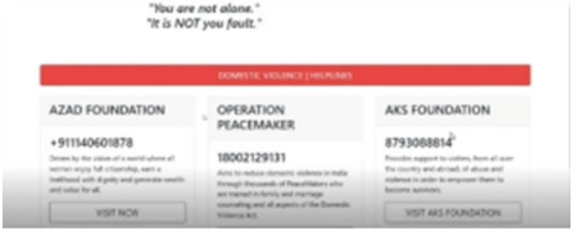
Tell us your story!

Text

He controlled what I wore, what I ate and drank, who I saw, who I spoke to, my social media, my telephone, our bank accounts, my credit car

Fig. 12 Section for the victim to write or narrate the story

Fig. 13 Output depicting the type of violence after the narration of story



"You are not alone."
"It is NOT you fault."

DOMESTIC VIOLENCE | HELPLINES

AZAD FOUNDATION
+911140601878
Driven by the vision of a world where all women enjoy full citizenship, earn a livelihood with dignity and generate wealth and value for all.
VISIT NOW

OPERATION PEACEMAKER
18002129131
Aims to reduce domestic violence in India through thousands of PeaceMakers who are trained in family and marriage counseling and all aspects of the Domestic Violence Act.

AKS FOUNDATION
8793088814
Provides support to victims, from all over the country and abroad, of abuse and violence in order to empower them to become survivors.
VISIT AKS FOUNDATION

and ANN with an accuracy of 79.89%. We then acquired a 98.50 percent accuracy after training the video dataset. Figure 14 depicts the suggested model's accuracy for each era. In addition, Fig. 15 shows the loss (Cross-Entropy loss) of both the train and test sets (Table 2)

.

For help hand signal detection, we got the accuracies as follows.

Hand segmentation, which is actually a tough operation in photographs with backgrounds containing different items, is not required with the suggested model.

Table 2 Performance evaluation for help hand signal detection

Help hand signal detection model	Accuracy
CNN	96.42
Inception V3	90
DNN with Squeezenet	83.28
ANN	79.89

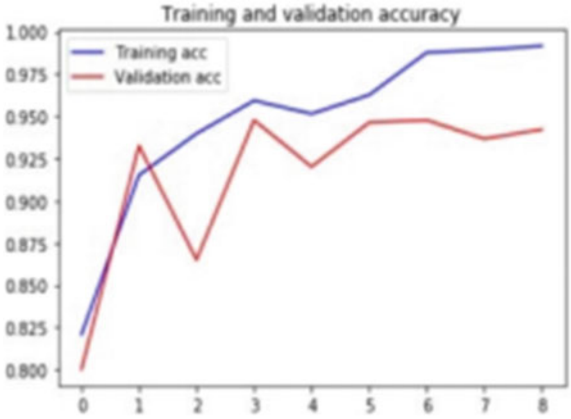


Fig. 14 Test versus training accuracy graph

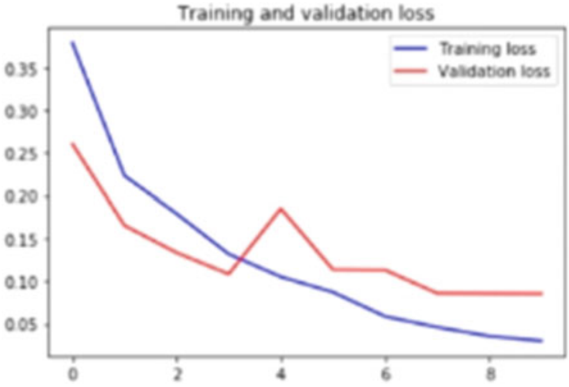


Fig. 15 Test versus training loss graph

Even though there are different segmentation approaches available based on skin color, hand shape, and various other factors, they all fail to produce accurate results when applied to photos with other background items or moving items. Our suggested method also removes the time-consuming task of identifying prospective feature descriptors capable of distinguishing different gesture types.

For multi-label classification, we got the accuracies as follows (Table 3).

We have employed domain-specific embeddings along with pre-trained embeddings. The analysis showed that domain-specific embeddings outperform pre-trained

Table 3 Performance evaluation for multi-label classification of stories

Multi-label classification model	Accuracy	Recall	Precision	F-measure	mean_pred
NN	0.921	0.861	0.833	0.879	0.385
CNN	0.926	0.865	0.856	0.88	0.397
RNN	0.914	0.829	0.819	0.861	0.393
CNN-Glove	0.912	0.805	0.755	0.847	0.418
RNN-Glove	0.908	0.788	0.797	0.831	0.433
CNN-Word2Vec	0.916	0.804	0.714	0.826	0.452
RNN-Word2Vec	0.912	0.825	0.798	0.825	0.47

embeddings. The graphs for training and validation accuracy along with the loss values are shown below for various models that we tested (Figs. 16, 17, 18 and 19).

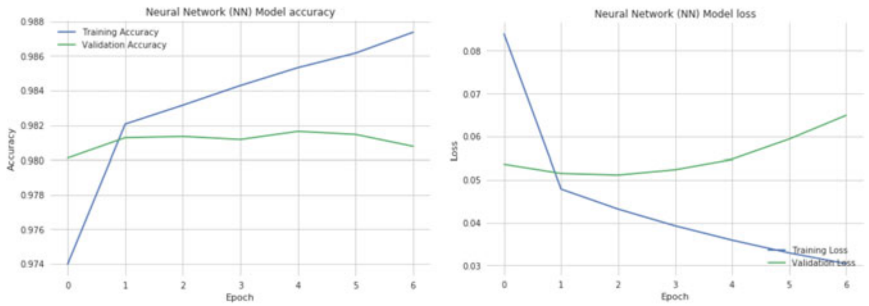


Fig. 16 Graph for the NN model

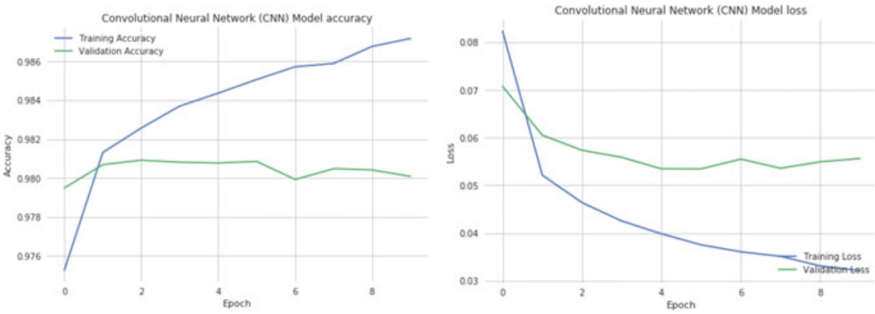


Fig. 17 Graph for the CNN model

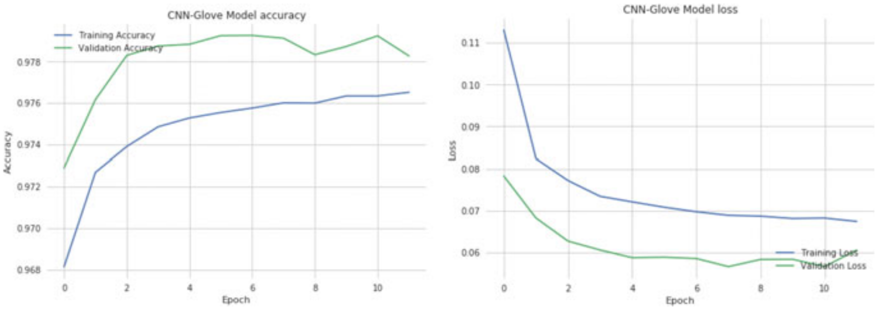


Fig. 18 Graph for the CNN-Glove model

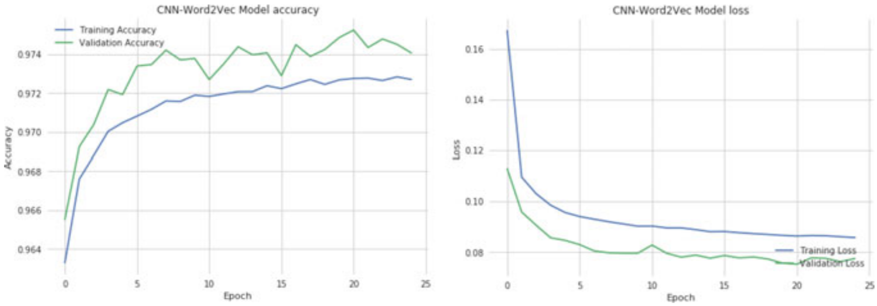


Fig. 19 Graph for the CNN-Word2Vec model

7 Conclusion

In many parts of the world, interpersonal violence, whether sexual or nonsexual, continues to be a big issue. It increases the victim's emotions of helplessness and impotence, lowering their self-esteem and indicating that they may be prone to additional violence [21]. In this project, we proposed a solution to combat violence and harassment with a violence detector based on NLP and Deep Learning methods. Nirbhaya Naari acts as a portal for women to raise their voices and firmly say no to violence. Our portal serves as a venue for receiving input by recording audio or video recordings of victims telling their stories or describing contact with violence. Alternatively, a victim can tell her tale in writing if she does not want to provide audio or video. Further, our system provides women with the function of "Help-Hand," which enables them to contact emergency contacts in case of an emergency and to communicate their whereabouts. In a nutshell, our project builds a platform where women can raise their voices and experience a safer environment. It will help to put terror in the minds of the abusers before they do any acts of violence. It also acts as an aid to ICC thereby helping to reduce the crime rate [22]. The model could be used as a web portal or in the future could be adapted to a mobile application for better functioning. Our work provides women the confidence to not just hold back but stand against their abusers fearlessly.

References

1. United Nations (1993) Declaration on the elimination of violence against women. UN, New York
2. Violence against women Prevalence Estimates (2018) Global, regional, and national prevalence estimates for intimate partner violence against women and global and regional prevalence estimates for non-partner sexual violence against women. WHO, Geneva, p 2021
3. https://wcd.nic.in/sites/default/files/Final%20Draft%20report%20BSS_0.pdf (NATIONAL SITE KA LINK)
4. Morrison A, Ellsberg M, Bott S (2007) Addressing gender-based violence: a critical review of interventions. World Bank Res Obs 22(1):25–51
5. Lutgendorf MA (2019) MD intimate partner violence and women's health. Obstet Gynecol 134(3):470–480. https://doi.org/10.1097/AOG.0000000000003326
6. Domestic violence screening and referral can be effective. Presented at the national conference on health care and intimate partner violence, San Francisco, CA, October 2000."
7. Di Franco M, Martines GF, Carpinteri G, Trovato G, Catalano D (2020) Domestic violence detection amid the Covid-19 Pandemic: the value of the who questionnaire in emergency medicine
8. Yut-Lin W, Othman S (2008) Early detection and prevention of domestic violence using the women abuse screening tool (Wast) in primary health care clinics in Malaysia. Asia Pac J Public Health 20(2):102–116. https://doi.org/10.1177/1010539507311899
9. Talboys S, Kaur M, Vanderslice J, Gren L, Bhattacharya H, Alder S (2017) What is eve teasing? A mixed methods study of sexual harassment of young women in the rural Indian context. SAGE Open 7:215824401769716. https://doi.org/10.1177/2158244017697168
10. Sharma M, Baghel R (2020) Video surveillance for violence detection using deep learning. In: Borah S, Emilia Balas V, Polkowski Z (eds) Advances in data science and management. Lecture notes on data engineering and communications technologies, vol 37. Springer, Singapore. https://doi.org/10.1007/978-981-15-0978-0_40
11. Wang P, Wang P, Fan E (2021) Violence detection and face recognition based on deep learning. Pattern Recognit Lett 142:20–24. ISSN 0167-8655
12. TY - BOOK AU - Wang, Jia-Ching AU - Wang, Jhing-Fa AU - Lin, Cai-Bei AU - Jian, Kun-Ting AU - Kuok, Wai-He PY - 2006/01/01 SP - 157 EP - 160 VL - 4DO. https://doi.org/10.1109/ICPR.2006.407
13. Shin J, Matsuoka A, Hasan MAM, Srizon AY (2021) American sign language alphabet recognition by extracting feature from hand pose estimation. Sensors (Basel) 21(17):5856. https://doi.org/10.3390/s21175856
14. Subramani S, Michalska S, Wang H, Du J, Zhang Y, Shakeel H (2019) Deep learning for multi-class identification from domestic violence online posts. IEEE Access 7:46210–46224. https://doi.org/10.1109/ACCESS.2019.2908827
15. Nievas EB, Suarez OD, Garc´ıa GB, Sukthankar R (2011) Violence detection in video using computer vision techniques. In: International conference on computer analysis of images and patterns. Springer
16. Hassner T, Itcher Y, Kliper-Gross O (2012) Violent flows: real-time detection of violent crowd behavior. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. https://doi.org/10.1109/cvprw.2012.6239348
17. Adithya V, Rajesh R (2020) A deep convolutional neural network approach for static hand gesture recognition. Procedia Comput Sci 171:2353–2361. ISSN 1877-0509

18. Koumoutsou D, Charou E (2020) A deep learning approach to hyperspectral image classification using an improved hybrid 3D-2D convolutional neural network. In: 11th Hellenic conference on artificial intelligence (SETN 2020). Association for Computing Machinery, New York, NY, USA, pp 85–92
19. Flach P (2019) Performance evaluation in machine learning: the good, the bad, the ugly, and the way forward. Proc AAAI Conf Artif Intell 33(01):9808–9814
20. Goutte C, Gaussier E (2005) A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In: European conference on information retrieval. Springer, Berlin
21. Kalra G, Bhugra D (2013) Sexual violence against women: understanding cross-cultural intersections. Indian J Psychiatry 55(3):244–249. https://doi.org/10.4103/0019-5545.117139
22. The International Criminal Court (ICC) https://www.government.nl/topics/international-peace-and-security/international-legal-order/the-international-criminal-court-icc

Pre-eclampsia Risk Factors Association with Cardiovascular Disease Prediction and Diagnosing Using Machine Learning

Ritu Aggarwal and Suneet Kumar

Abstract Preeclampsia disease is a kind of disorder which usually occurs due to high blood pressure during pregnancy in women. It includes cardio changes, protein less intake, abnormalities in hematologic or some cerebral, urine manifestations. This disease effects 3–5% at the time of pregnancy. It is the reason for preeclampsia symptoms. In this proposed work machine learning techniques are applying to improve the prediction rate and diagnosis, prevention of complex disease with their symptoms. Most of women have affects a risk of cardiovascular disease due to preeclampsia. It affects the heart of women or also affects the other organs of the baby and mother. The aim of the study is to propose the prediction model by selecting features of particular class using the dataset. The dataset has 303 instances and 14 attributes for cleveland which is used for heart disease and 1550 samples and 30 features taken by collecting with clinical serum for the samples of 1000 out of 1550 for preeclampsia women. These samples are divided according to training and testing ratio of 7:3. Prediction based model is developed for implementing through machine learning it is used for prenatal CVD threat in PC suffered women.

Keywords Cardiovascular disease · Preeclampsia · Machine learning · Blood pressure

1 Introduction

CVD is the main reason of death all over the world. The ratio of women who are suffering from this disease. The death rate in women has expanded due to preeclampsia. It is a hypersensitive issue during pregnancy. The common factors

R. Aggarwal (✉)
Deparment of Computer Science and Engineering, Maharishi Markandeshwar Engineering College, Mullana, Haryana, India
e-mail: errituaggarwal@gmail.com

S. Kumar
Maharishi Markandeshwar Engineering College, Mullana, Ambala, Haryana, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*, Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_5

related to CVD and preeclampsia such as hypertension, high blood pressure, digestion, change in veins, high protein in blood, etc. (6, 7). Pregnancy is related to some changes that are pathologic or physical related. If the pregnancy issue is related to pathologic it means the condition is preeclampsia. The term used is related to preeclampsia that is eclampsia a condition that shows the cardiovascular changes in the patient body. Due to this some other kind of disease could be occurred such as hematologic abnormalities, hepatic, neurologic, and others [1]. It is a scientific disorder that burdens 3–5% of pregnancies and is a main source of maternal mortality, particularly in agricultural nations. It raises serious preeclampsia hem dialysis and liver-related diseases that all are comes under hypertensive tissues and low platelets syndrome [2]. Preeclampsia is characterized as the new beginning of hypertension and proteinuria during the last part of pregnancy that ordinarily shows up around 20 weeks of incubation with manifestations of hypertension and proteinuria. Significant fringe vasoconstriction alongside diminished blood vessel consistence prompting uncontrolled hypertension has been accounted for [3, 4]. A couple of clinical conditions, increment the danger of Preeclampsia: primiparity, past preeclamptic pregnancy, persistent hypertension or constant renal sickness, or both. An danger with toxemia is expanded twofold to fourfold assuming a patient has a first-degree relative with a clinical history of the issue and is expanded sevenfold if toxemia is muddled a past pregnancy. Numerous incubations are an extra danger factor; trio growth is a more serious danger than twin development. Trademark cardiovascular danger factors likewise are related to an expanded likelihood of toxemia, as are maternal age more established than 40 years, diabetes, corpulence, and previous hypertension. The expanded commonness of constant hypertension and other comorbid clinical sicknesses in ladies more seasoned than 35 years might clarify the expanded recurrence of toxemia among more established ladies. Customarily, the conclusion of PE relies straightforwardly upon the well-being proficient. This finding can be improved with the utilization of e-well-being strategies. These can uphold the anticipation of the infection, keeping away from the issues that happen whenever it has been analyzed. The flow focal point of medical care analysts is to advance the utilization of well-being innovation in non-industrial nations to help clinical decisions [5]. Notwithstanding the clinical history and family ancestry, it is critical to consider the financial variables wherein ladies are submerged during pregnancy. Low Socio-financial elements go about as various danger factors for toxemia. They are related to dietary issues, decreased risk natal consideration, and unsanitary clean conditions. In Mexico, low financial status of women multiplied the danger of preeclampsia and CVD [6]. According to previous reviewers observed that working ladies contrasted with non-working ones had a higher danger of creating preeclampsia and eclampsia [7]. This might be connected to the pressure that ladies get during work. Most of the tools which are used for prediction could detect the people's diseases or which may have a chance of high disease infection so that it could give benefits to that patient who is exactly suffered from CVD. It is an advantage to detect the disease at its early stages [4, 8–10]. According to ACC/AHA find a research in young ladies there is a higher chance of CVD as measured by the different tools. In young ladies during pregnancy if they suffered from CVD there is a chance the child is get effected by disease. So that in each patient early detection and diagnosis of the disease is a major requirement for toxemia with a background marked by toxemia [11–13]. To this end, a checked apparatus is direly needed to screen out high-hazard preeclampsia women with post-pregnancy CVD and play out a designated mediation. On their risk factors, some of the devices is used to detect the CVD before toxemia. The different machine learning techniques were used to incorporate numerous factors to achieve accuracy in prediction of results. According to the research and different studies preeclampsia and CVD are portrayed to be danger. This paper analyzes for pregnant women to endure preeclampsia, and presents a few manners by which the choices made by the framework are reasonable to trained professionals. The following paper is as follows: in Sect. 1 introduction is discussed, in Sect. 2 related works in which existing studies about these diseases will be discussed by researchers. Section 3 proposed works using machine learning with preeclampsia. Section 5 discussions and results based on preeclampsia with cardiovascular disease. Section 6 conclusion of work discussed with future scope.

1.1 Research Gap and Objective

İn the previous work none worked on a real dataset. The preeclampsia disease is directly related to the heart disease. No one discusses it with heart disease. At the stage of pregancy the heart disease is occurring because of high blood pressure, hypertension. İn this current work improved technologies and models according to existing work done by researchers. The different attributes of HD and preeclampsia that show their relationship.

2 Literature Review

Sufriyana et al. [1] in this study proposed a model using the BPJS dataset which was implemented to detect the preeclampsia in pregnant women with different machine learning algorithms using performance metrics. Li et al. [2] in this researcher proposed a early identification of patients at a risk of PE. The results were obtained by different parameters. AUC obtained 0.92 at the highest point. Espinilla et al. [5] in this study researcher proposed a study to diagnose and support the disease. The methodology is based on the fuzzy linguistic approach in which the data extraction process is composed of two phases. Sonnenschein et al. [6] proposed a work for the detection of preeclampsia at early stages using the dataset of PAD. The results were implemented using artificial intelligence by applying machine learning. Random forest gives an evaluation matrix by using some parameters. de Havenon et al. [7] in this proposed work the researcher studies the risk factor against the preeclampsia disease. Its implementing on the time varying vascular risk factors. Wang g et al. [14] proposed a study using machine learning algorithms. It has taken the dataset of

CVD and preeclampsia. For training and testing the tenfold cross-validation test set was evaluated. Random forest algorithm is considered as a good approach for the prediction of disease. It is also calculated systolic pressure.

Lee et al. [15] proposed a work related to showing the key information for disease prediction by adding the features. It proposed a approach to calculate the mean and median that choose individual properties of dataset. Steinthors dottir et al. [16] in this study the researcher proposed a work related to the maternal genome with their characteristics in which at the time of birth the disease is detected and diagnosed. The hypertension score is calculated which shows how many effects pregnant ladies at the time of birth of child. It computed the results of GWAS for using the five variants that associated the risk factors of disease. As concluded that the hypertension is a major risk factor in preeclampsia. Melton [17] in this study projected a WES approach for detecting the preeclampsia with two novel genes that describe the novel feature of preeclampsia. It's used technology ANXA5. The use of this technology solves the complications regarding the disease at its early stages.

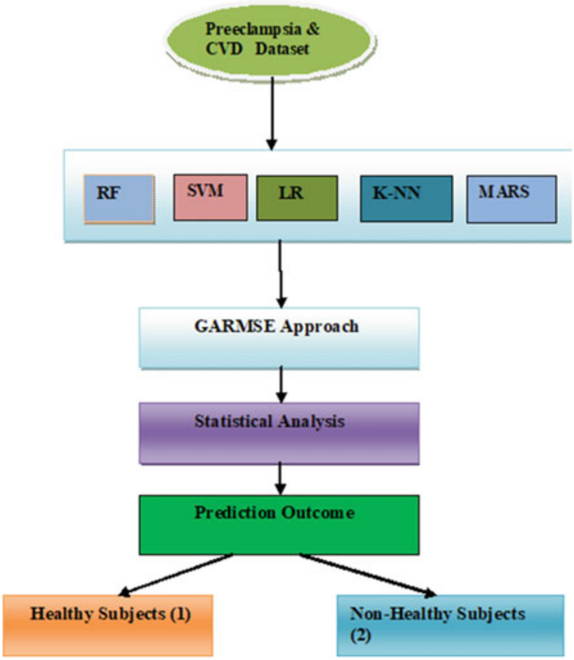
3 Proposed Work with Dataset and Tools Used

In this proposed study the preeclampsia is implemented with ML methods to improve the medical techniques and get accuracy in results. In the health care systems present results using the machine learning classifiers so that medical physicians understand how to diagnose and predict the results of the disease. GARMSE method of machine learning is used to compute the accuracy, precision, Recall, Score, etc. These are the performance which is used to calculate the results using approaches of ML.

3.1 Dataset and Methods, Tools

The dataset will be taken from the child health care development from the hospital of Apollo Cradle Hospital, Gurugram—Best Maternity and New-born Care. The dataset has 1550 samples and 30 features from these samples. This dataset has a labeled features of age, sex, order, birth weight, month of pregnancy, when pregnancy starts, number of antenatal visit, sonography, risk factors during pregnancy, obesity measurements, etc., as shown by Fig. 1 and the pseudo-code for GARMSE.

Fig. 1 Proposed work flow



Pseudo code for GARMSE:

```
1. Initialization of population
2. Evaluate the population
3. When the generation = 0
4. Then select the features of preeclampsia and CVD using RMSE
5. If the mean value < = 0.9
6. Then initialize generation = generation + 1
7. After then prioritize the features which show disease by setting their value to 1, 0
8. Mean = mean + 1
```

The above pseudocode describes the workflow of GARMSE which firstly chooses the population at then evaluates it according to the mentioned attributes. After that put the value of generation as 0 and apply the RMSE to compute the error in the dataset. After that select only that attribute from the entire population which is relevant.

4 Results and Discussions

The following conditions check for Preeclampsia is given before resulting the outcomes:-

4.1 Mild PE

The symptoms for Mild PE are counted BP as 140/90 mmHg during pregnancy and are greater than 350 mg. The maximum time for this is 24 h. The volume is as 500 ml [16, 18, 19].

4.2 Severe PE

If the BP of pregnant women is 165/100 mmHg as SP greater than 60 mmHg and the DBP is <30 mm Hg in 24 h and after the 20th week the most common symptoms are insights such as edema, headache, hearing and vision disturbances, pain in the right headache, visual disturbances, pain in the right hypochondriac high level of abnormal liver enzymes, acute fatty liver, etc. [15, 17, 20].

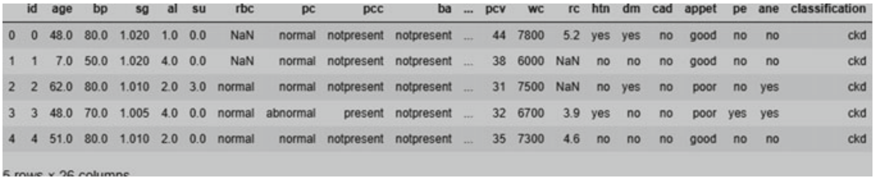
In this study prediction and diagnosis are based on the mild or severe PE. The probability of predicting the disease will be find out by the machine learning classification models. The machine learning classifiers RF, SVM, LR, and MARS are used to learn the relations between different classes of prediction values.

In the first step the dataset is preprocessed which has some missing values and imbalanced features are to be extracted from the dataset. In the second step used type prediction classification methods on sample dataset [21]. GARMSE algorithms are used to analyzing the ROC graph with evaluation metrics. In the last step a sample of 1550 out of which 30 features are selected and their statistical analysis carried out by the computing for healthy and non-healthy subjects. In the samples 1550 out of that 280 are analyzed as suffering from Preeclampsia and other 1270 are healthy.

The columns are labeled with numeric value and categorical value. Preeclampsia is predicted and labeled as 1 healthy subject and Unhealthy subject as 2. The best model GARMSE is considered for this work which gives better outcome in terms of accuracy. This model is best out performed in their performance evaluation metrics for all validation sets.

4.3 Garmse

It is based on the multiparameter tuning of hyperparameters. It is based on the heuristic search which solves searching and optimization problems [22, 23]. Basically GARMSE measures the prediction error in results and measures the flow of regression through the data points with the regression line. It computes the mean value with the fitness function. In this work GARMSE gives the best results for predicting the disease's presence and absence. As shown in Fig. 2 the dataset with attributes shows their attribute values as per subject mentioned. The GARMSE results are different from other approaches and models which were used by the other researchers because



```
   id   age    bp     sg   al   su     rbc        pc         pcc          ba ... pcv   wc   rc  htn  dm  cad appet   pe  ane classification
0   0  48.0  80.0  1.020  1.0  0.0     NaN    normal  notpresent  notpresent ...  44 7800  5.2  yes yes   no  good   no   no            ckd
1   1   7.0  50.0  1.020  4.0  0.0     NaN    normal  notpresent  notpresent ...  38 6000  NaN   no  no   no  good   no   no            ckd
2   2  62.0  80.0  1.010  2.0  3.0  normal    normal  notpresent  notpresent ...  31 7500  NaN   no yes   no  poor   no  yes            ckd
3   3  48.0  70.0  1.005  4.0  0.0  normal  abnormal     present  notpresent ...  32 6700  3.9  yes  no   no  poor  yes  yes            ckd
4   4  51.0  80.0  1.010  2.0  0.0  normal    normal  notpresent  notpresent ...  35 7300  4.6   no  no   no  good   no   no            ckd

5 rows x 26 columns
```

Fig. 2 Dataset results showing categorical values

Table 1 Results for Preeclampsia for 30 features and 1550 samples

Classifier	ROC	Accuracy	Avg Accuracy	Precision	Recall	Kappa	F score	Time
RF	0.8259	0.823	0.761	0.6976	0.8114	0.1862	0.3166	209.461
MAR	0.8011	0.8011	0.726	0.5632	0.8327	0.3113	0.442	7.656
SVM	0.7781	0.834	0.7142	0.5784	0.879	0.2756	0.373	13.879
KNN	0.7775	0.8289	0.6945	0.5154	0.8805	0.2967	0.3577	42.521
Logistic Regression	0.7988	0.8286	0.6955	0.5323	0.8895	0.3558	0.4858	7.368
GARMSE	0.8519	0.873	0.799	0.6354	0.8976	0.4125	0.4776	39.61

as per the related disease no one implemented this model for their work. It is a new technique which obtained an accuracy of 0.873 that is much better than the other work. This approach easily predicts the disease at its early stages.

The results obtained by the given Table 1 using different classifier used to predict the disease preeclampsia, the best outcome obtained by the GARMSE method 0.873 accuracy and ROC as 85.19 as shown by Box 1 algorithm for preeclampsia.

The given model shows predicted samples with values close to 1 and their implementation with performance metrics for individually as 0.85 which are health subjects. The given Fig. 3 is showing ROC characteristics for the best approach that give better results prediction that is GARMSE. The dataset shows categorical values by putting the attributes. The roc results are based on a confusion matrix by which all classifiers of machine learning are tested and trained. According to those results the AUC results in terms of prediction outcome is obtained as 85.19. (Box 1)

5 Conclusions and Future Work

This proposed study implements different classifiers to build up a model using the GARMSE approach that individually detects who suffers from preeclampsia disease. The different subjects employed who are healthy and unhealthy which is labeled as 1 and 2. In this at the initial stage trying to detect the disease during pregnancy. AUC characteristics with GARMSE gives better outcome such as 8519 and accuracy

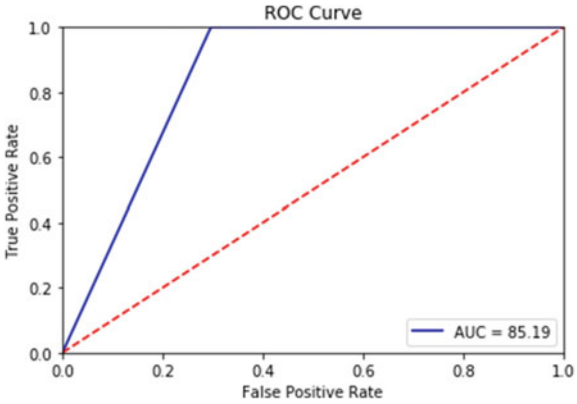


Fig. 3 ROC characteristics

If the age of women<=30 that is mestizo

Time of pregnancy = 3T

BMI <=29 then the risk of preeclampsia then the risk of pregnancy =2T

Box 1 Algorithm steps for preeclampsia

achieved 0.873 at a time interval of 39.61, respectively. In future extends this work by choosing more samples according to demographic details and feature extraction applying with deep learning models. So that in early stages give more accuracy and prediction.

References

1. Sufriyana H, Wu YW, Su ECY (2020) Artificial intelligence-assisted prediction of preeclampsia: development and external validation of a nationwide health insurance dataset of the BPJS Kesehatan in Indonesia. EBioMedicine 54:102710
2. Li S, Wang Z, Vieira LA, Zheutlin AB, Ru B, Schadt E, Li L (2021) Improving pre-eclampsia risk prediction by modeling individualized pregnancy trajectories derived from routinely collected electronic medical record data. medRxiv
3. Wang G, Zhang Y, Li S, Zhang J, Jiang D, Li X, Du J (2021) A machine learning-based prediction model for cardiovascular risk in women with preeclampsia. Front Cardiovasc Med 1465

4. Aggarwal R, Kumar S (2022) Nomenclature of machine learning algorithms and their applications. Data science for effective healthcare systems 161–168
5. Espinilla M, Medina J, García-Fernández ÁL, Campaña S, Londoño J (2017) Fuzzy intelligent system for patients with preeclampsia in wearable devices. Mob Inf Syst
6. Sonnenschein K, Stojanović SD, Dickel N, Fiedler J, Bauersachs J, Thum T, Tongers J (2021) Artificial intelligence identifies an urgent need for peripheral vascular intervention by multiplexing standard clinical parameters. Biomedicines 9(10):1456
7. Hammoud GM, Ibdah JA (2014) Preeclampsia-induced liver dysfunction, HELLP syndrome, and acute fatty liver of pregnancy. Clin Liver Dis 4(3):69
8. Aggarwal R, Kumar S (2022) An automated perception and prediction of heart disease based on machine learning. In: AIP conference proceedings (vol 2424, No. 1, p 020001). AIP Publishing LLC.
9. Brown MC (2013) Cardiovascular disease risk in women with pre-eclampsia: systematic review and meta-analysis. Eur J Epidemiol 28:1–19
10. Brouwers L (2018) Recurrence of pre-eclampsia and the risk of future hypertension and cardiovascular disease: a systematic review and metaanalysis. BJOG An Int J Obstet Gynaecol 125:1642–1654
11. Evangelou E et al Genetic analysis of over 1 million people identifies 535 new loci associated with blood pressure traits. Nat Genet 50:1412–1425
12. Beaumont RN (2018) Genome-wide association study of offspring birth weight in 86 577 women identifies five novel loci and highlights maternal genetic effects that are independent of fetal genetics. Hum Mol Genet 27:742–756
13. Yang J, Lee SH, Goddard ME, Visscher PM (2011) GCTA a tool for genome-wide complex trait analysis. Am J Hum Genet 88:76–82
14. de Havenon A, Delic A, Stulberg E, Sheibani N, Stoddard G, Hanson H, Theilen L (2021) Association of preeclampsia with incident stroke in later life among women in the Framingham heart study. JAMA Netw Open 4(4):e215077–e215077
15. Lee TE (2017) Predicting key features of a substation without monitoring. Math-In-Ind Case Stud 8(1):1–9
16. Steinthorsdottir V, McGinnis R, Williams NO, Stefansdottir L, Thorleifsson G, Shooter S, Morgan L (2020) Genetic predisposition to hypertension is associated with preeclampsia in European and Central Asian women. Nat Commun 11(1):1–14
17. Melton PE (2019) Whole-exome sequencing in multiplex preeclampsia families identifies novel candidate susceptibility genes. J Hypertens 37(997–1011):10
18. Hansen AT, Jensen JMB, Hvas AM, Christiansen M (2018) The genetic component of preeclampsia: a whole-exome sequencing study. PLoS ONE. https://doi.org/10.1371/journal.pone.0197217
19. Feitosa MF (2018) Novel genetic associations for blood pressure identified via gene-alcohol interaction in up to 570K individuals across multiple ancestries. PLoS ONE 13:e0198166
20. Aggarwal R, Podder P, Khamparia A (2022) ECG classification and analysis for heart disease prediction using XAI-driven machine learning algorithms. In: Biomedical Data Analysis and Processing Using Explainable (XAI) and Responsive Artificial Intelligence (RAI). Intelligent systems reference library, vol 222. Springer, Singapore. https://doi.org/10.1007/978-981-19-1476-8_7
21. Aggarwal R, Kumar S (2022) HRV based feature selection for congestive heart failure and normal sinus rhythm for meticulous presaging of heart disease using machine learning. Measurement: Sensors 24:100573
22. Sung YJ (2018) A large-scale multi-ancestry genome-wide Study accounting for smoking behavior identifies multiple significant loci for blood pressure. Am J Hum Genet 102:375–400
23. Frayling TM (2018) A common variant in the FTO gene is associated with body mass index and predisposes to childhood and adult obesity. Science 316:889–894

A Low Resource Machine Learning Approach for Prediction of Dressler Syndrome

Diganta Sengupta, Subhash Mondal, Debosmita Chatterjee, Susmita Pradhan, and Pretha Sur

Abstract Cosmopolitan lifestyle and livelihood modifications have marked a toll on human health to the extent of myocardial disease onset at a relatively tender stage. One of the major issues that have been observed on the rise is the arterial blockage leading to myocardial infarction. Immune response to the arterial damage or the pericardium is termed as Dressler syndrome. This study focuses on prediction of Dressler syndrome based on myocardial infarction historical data. Moreover, the study focuses on prediction using a resource constraint dataset through six popular machine learning (ML) algorithms. The dataset comprised of 124 features, and 1700 data, post-cleaning. Of all the 124 features, 12 features were target values. We selected one of the target values (Dressler syndrome) for this study. 10% of the data was reserved for test data at the initial stage itself, and the rest was further split into 0.7:0.3 for training and validation sets. RF presented a model accuracy of 98%, which is the best of all the six algorithms. In terms of AUC, RF exhibited the highest value of 0.995. Moreover, the models were further tuned, and the results confirmed the efficacy of RF for the classification of Dressler syndrome.

D. Sengupta (✉) · S. Mondal · D. Chatterjee · S. Pradhan · P. Sur
Department of Computer Science and Engineering, Meghnad Saha Institute of Technology, Kolkata 700150, India
e-mail: sg.diganta@ieee.org

S. Mondal
e-mail: subhash@msit.edu.in

D. Chatterjee
e-mail: debosmita_c.cse2019@msit.edu.in

S. Pradhan
e-mail: susmita_p.cse2019@msit.edu.in

P. Sur
e-mail: pretha_s.cse2019@msit.edu.in

D. Sengupta
Department of Computer Science and Business Systems, Meghnad Saha Institute of Technology, Kolkata 700150, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_6

Keywords Myocardial infarction · Dressler syndrome · Machine learning · Artificial intelligence · Bioinformatics

1 Introduction

Myocardial infarction popularly known as heart attack or cardiac arrest accounts for over one-fourth of the present annual global fatality [1]. Clinically it has been proven that the process initiates with a decline of blood inflow to the heart muscles. Multiple reasons have been cited till date for the decline such as arterial blockage due to cholesterol sedimentation in the arteries, excessive alcohol intake followed by poor diet, excessive stress, blood clotting, and in some cases cellular waste leading to the blood clot [2]. Present work stress followed by changing socio-economic lifestyle has aided in the growth of the decline parameters leading to the major share of fatalities through myocardial infarction. Post-myocardial infarction, the human immune system tries to initiate self-healing measures against the trauma caused to the heart muscles. This leads to inflammation of the membrane that encapsulates the heart (pericardium). This inflammation is clinically termed as pericarditis which is a common symptom of post-myocardial infarction. Another common symptom is the swelling of the pleurae leading to immense pain (pleuritic pain), and fever. All these symptoms taken together are clinically termed as Dressler syndrome. It has been observed that Dressler syndrome generally results from heart surgery, chest trauma, and myocardial infarction. Also it has been seen that the syndrome affects an age bracket of 20–50 years [3]. Also it has been observed that owing to a wide range of clinical presentations is usually tough for health professionals to recognize.

Dressler Syndrome being an immune system reaction may also lead to fluid build-up in the surrounding tissues of lungs also known as pleural effusion [4]. The build-up can put pressure on the heart muscles compelling them to work hard [4]. Chronic pathological inflammation can cause the pericardium to become scarred or thick, because of this heart's inability to efficiently pump blood [4]. So, it becomes important to timely identify Dressler Syndrome to minimize further risks for the patients. This served as the motivation for the study. We classify Dressler syndrome using Machine Learning (ML) algorithms based on historical data related to myocardial infarction leading to Dressler syndrome. Thereafter we claim that if Dressler syndrome is observed in a patient, then either the patient has had a myocardial infarction or is going to experience myocardial infarction.

We have chosen ML algorithms for this study because the dataset is resource constraint [5] which contains approximately a total of 1700 samples, the details of which are presented further in the paper. Multiple approaches have been applied to extract the best way of determining whether post-myocardial infarction, Dressler syndrome can occur or not. We present only the best results in this paper, excluding the other approaches we did which resulted in lower performance metrics results. Six ML algorithms have been used in this study as follows: Random Forest (RF), Xtreme Gradient Boost (XGB), Support Vector Machine (SVM), Decision Tree (DT), K Nearest Neighbor (KNN), and Logistic Regression (LR). The performance metrics which serve as the parameter of evaluation for the ML algorithms are accuracy, recall, precision, F1-score, and AUC score.

To the best of our knowledge, this is the first study which proposes a binary classifier model based on conventional ML algorithms which presents whether a patient has experienced or is going to experience myocardial infarction, based on a diagnosis of Dressler syndrome.

The rest of the paper is organized as follows. The next section presents the related work with respect to this study followed by the proposed classification models including the data preprocessing techniques presented in Sect. 3. The results are presented in Sect. 4 followed by the Discussion and Conclusion in Sect. 5.

2 Related Work

Although the study in this paper is novel, we present a few related works of importance in terms of myocardial infarction and Dressler syndrome. Authors in [1] have used ECG (electrocardiogram) signals for the prediction of myocardial infarction. The ECG signals have been decomposed in wavelets, thereby generating different clinical components within different sub-bands of the wavelets which are captured by Eigen space-based features, and wavelet entropy. In that study, KNN evolved as the best classifier seconded by SVM. They also presented a comparative analysis with convolutional neural networks. Their study focused on the prediction of myocardial infarction through wavelet decomposition of ECG signals using ML algorithms. The use of ECG signals for the prediction of myocardial infarction has been further proposed in [2, 6], using ML algorithms, in [7, 8] using DL algorithms. Authors in [8] also used Recurrent Neural Networks (RNN) for the prediction. Low-quality ECG signals have been used for the early detection of myocardial infarction in [9]. The authors have used DL frameworks for detection. Another approach for detection of myocardial infarction using DL algorithms is presented in [10] where the authors provide a two-fold approach, one class-based approach and another subject-based approach. Other ML-based approaches can be found in [11, 12].

Another approach for the prediction of myocardial infarction using ML algorithms is presented in [13]. The authors used a resource-constrained dataset containing a feature set of 26, and 345 instances. Three classes were presented in the dataset, namely Distinctive, Non-distinctive, and both (Distinctive and Non-Distinctive). Basically this study focused on multi-class prediction of myocardial infarction using ML algorithms such as Bagging, LR, and RF. The authors claimed accuracies of 93.91%, 93.63%, and 91.02%, respectively, for the three ML algorithms. Authors in [14] also predicted myocardial infarction using ECG signals, in which they generated a feature set containing twenty-one time domain features which had been extracted from ECG signals. This study focused on the use of Deep Learning (DL) algorithms such as Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNN). Their results exhibited training and testing accuracy of 99.05%, and 98.50%, respectively, using CNN and Bidirectional LSTM. Another noted work using LSTM can be found in [15]. Authors in [16] have done a comparative analysis of the detection of myocardial infarction using ML and DL algorithms. They used SVM in one study and Artificial Neural Networks (ANN) in another. They claim that SVM fared better than the DL counterpart.

3 Proposed Work

In this section we present the proposed workflow used for classification of Dressler Syndrome. Initially the data analysis comprised of three parts as discussed in Sect. 3.1 through Sect. 3.2. Then the model was trained using the ML algorithms. The workflow is presented in Fig. 1.

3.1 Dataset Acquisition

This present study was conducted using the dataset from [5] which comprised of 1700 instances and 124 features. Of the 124 features, the first 111 features ranging from column 2 to column 112 are input features for classification or prediction. The rest 12 columns from column 113 to column 124 contain target labels which denote complications that can arise from myocardial infarction. The dataset is a recent dataset and can be used for both classification as well as prediction. In this study only one of those 12 target labels has been used the Dressler syndrome. The dataset contains missing values which have been handled using data preprocessing as discussed in the subsequent subsections.

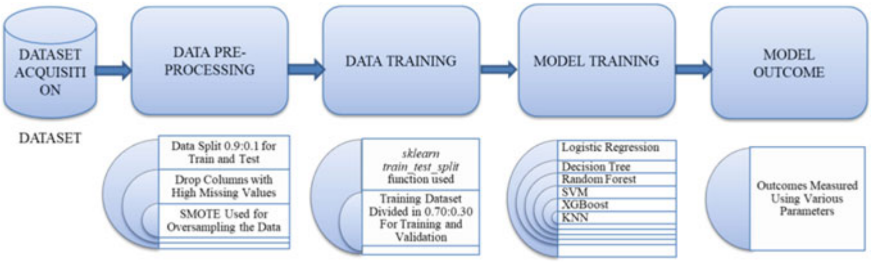

DATASET ACQUISITI ON
DATA PRE-PROCESSING
DATA TRAINING
MODEL TRAINING
MODEL OUTCOME
DATASET
Data Split 0.9:0.1 for Train and Test
Drop Columns with High Missing Values
SMOTE Used for Oversampling the Data
sklearn train_test_split function used
Training Dataset Divided in 0.70:0.30 For Training and Validation
Logistic Regression
Decision Tree
Random Forest
SVM
XGBoost
KNN
Outcomes Measured Using Various Parameters

Fig. 1 Proposed workflow

3.2 Data Pre-processing

The dataset was split into two parts randomly to generate the test and the train set. As discussed earlier, a number of approaches had been used for splitting. We had used cross-validation to finalize the train test split ratio. Finally we focused on a split ratio of train to test as 0.9:0.1. Hence 10% of the dataset was used for testing, and 90% of the data was used for training as well as validation purpose. Column number 120 contained the Dressler syndrome. Hence barring column 120, we dropped the other 11 label columns from the study. Due to the high percentage of missing values of a particular column, we have also dropped one input column labeled IBS_NASL. Although other feature columns too contained missing values, but their count being admissible, we retained those features and handled them. Also it may be noted that the dataset is highly imbalanced containing 1462, and 68 values for the binary classes of 0, and 1, respectively. Hence, this class imbalance was also handled using the popular oversampling technique called SMOTE (Synthetic Minority Oversampling Technique). The application of SMOTE technique resulted in oversampled instances of 1462 values for each of the two classes, respectively.

3.3 Data/Model Training

For training the model, as discussed earlier, six ML algorithms were used. Initially the training dataset was split into a train and validate dataset using a ratio of 0.7:0.3, respectively. The training of the models was done using a popular ML library *sklearn* [17]. The ratio for 0.7:0.3 was again obtained using cross-validation in a random manner. The complete dataset comprised of 1700 instances. As 10% (170) of the instances were used as testing, the remaining 1530 instances were used for training and validation having the count of 1071 and 459, respectively. The choice of the six ML algorithms was based on prior art which contained prediction, and classification of myocardial infarction as discussed in the Related Work section. Out of the existing literature, the top six best performing algorithms in terms of the performance metrics were chosen for the study. The performance of the algorithms can be obtained from the related papers cited in the Related Work section.

4 Result Analysis

This section presents the results obtained from the ML algorithms in terms of their performance on the processed dataset. As discussed earlier, five performance metrics have been used to evaluate the performances. Table 1 presents the results thus obtained.

The results from Table 1 are graphically presented in Figs. 2, 3, 4, 5, and 6, respectively. It can be observed that although RF exhibits the best results in terms of all the performance metrics. Even the F1-Score for RF stands the best which is further established through the ROC-AUC score.

The ML models were further trained using the hyper-parameter tuned values. Tables 2 and 3 present the performance values with respect to the tuned versions. Table 2 presents the results with respect to *RandomisedSearchCV*, and Table 3 presents with respect to *GridSearchCV*. The tuned study was done to further validate the decision that RF generates the best classification result.

Table 1 Performance metric analysis for the respective machine learning models

Model	Accuracy (%)	Precision	Recall	F1-score	ROC-AUC score
LR	0.82	0.86	0.87	0.81	0.99
DT	0.92	0.94	0.94	0.92	0.993
RF	0.97	1	1	0.97	0.98
SVM	0.97	0.95	1	0.97	1
XGB	0.97	1	0.94	0.97	0.98
KNN	0.83	0.75	1	0.85	0.97

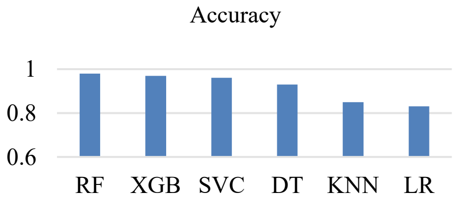


Fig. 2 Performance evaluation of six ML algorithms in terms of accuracy

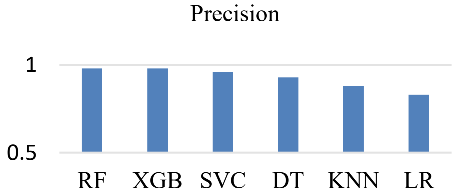


Fig. 3 Performance evaluation of six ML algorithms in terms of precision

Fig. 4 Performance evaluation of six ML algorithms in terms of recall

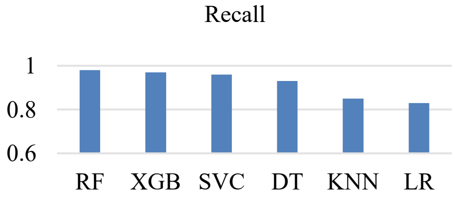


Fig. 5 Performance evaluation of six ML algorithms in terms of F1-score

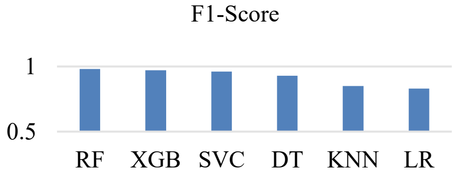


Fig. 6 Performance evaluation of six ML algorithms in terms of ROC-AOC score

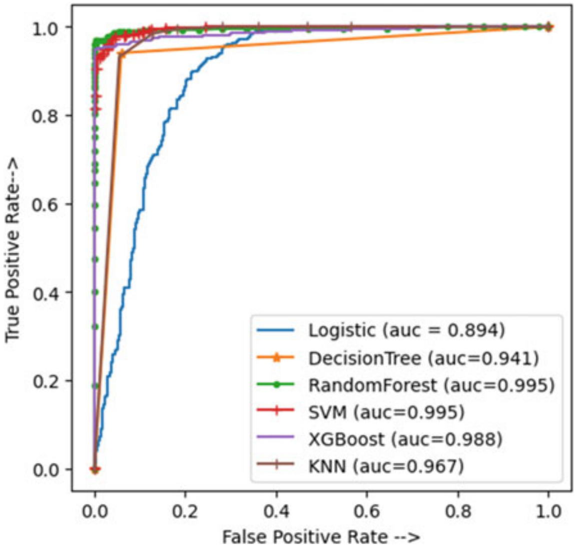


Table 2 Performance metric analysis for the tuned models with *RandomisedSearchCV*

Model	Accuracy (%)	Precision	Recall	F1-score	ROC-AUC score
LR	0.84	0.86	0.9	0.83	0.9
DT	0.95	0.94	0.93	0.92	0.93
RF	0.98	1	0.95	0.975	0.96
SVM	0.85	0.81	0.93	0.86	0.85
XGB	0.97	0.996	0.95	0.97	0.97
KNN	0.86	0.79	1	0.85	0.96

Table 3 Performance metric analysis for the tuned models with *GridSearchCV*

Model	Accuracy (%)	Precision	Recall	F1-score	ROC-AUC score
LR	0.84	0.86	0.89	0.83	0.9
DT	0.93	0.94	0.94	0.92	0.93
RF	0.98	1	0.95	0.975	0.98
SVM	0.85	0.81	0.93	0.87	0.85
XGB	0.97	0.987	0.95	0.97	0.97
KNN	0.86	0.79	1	0.85	0.96

Table 4 presents the code for the tuned values with respect to the two tuning algorithms. Figure 7 presents the confusion matrices for the usual implementation of the models for the ML models. Figures 8 and 9 present the confusion matrices for the tuned models using *RandomisedSearchCV* and *GridSearchCV*.

Table 4 Hyper-parameter values for the two tuning algorithms

Model	RandomizedSearchCV	GridSearchCV
Logistic regression	{'class_weight': 'balanced', 'dual': False, 'max_iter': 250, 'penalty': 'l2'}	{'class_weight': 'None', 'dual': False, 'max_iter': 250, 'penalty': 'none'}
Decision tree classifier	{'criterion': 'entropy', 'max_depth': 560, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2}	{'criterion': 'entropy', 'max_depth': 560, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2}
Random forest classifier	{'criterion': 'gini', 'max_depth': 230, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1400}	{'bootstrap': True, 'max_depth': None, 'max_features': 'auto', 'n_estimators': 11}
SVM	{'C': 1000, 'degree': 3}	{'C': 1000, 'degree': 5, 'kernel': 'poly'}
XGBOOST	{'colsample_bylevel': 0.7, 'colsample_bytree': 0.8, 'gamma': 0, 'learning_rate': 0.2, 'max_depth': 15, 'min_child_weight': 0.5, 'n_estimators': 100, 'reg_lambda': 1.0, 'silent': False, 'subsample': 0.5}	{'colsample_bytree': 0.5, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 7, 'reg_lambda': 10, 'scale_pos_weight': 3, 'subsample': 0.8}
ADABOOST	{'learning_rate': 1.0, 'n_estimators': 50}	{'learning_rate': 0.1, 'n_estimators': 500}
GRADIENTBOOST	{'learning_rate': 0.15, 'n_estimators': 1500}	{'learning_rate': 0.05, 'n_estimators': 250}

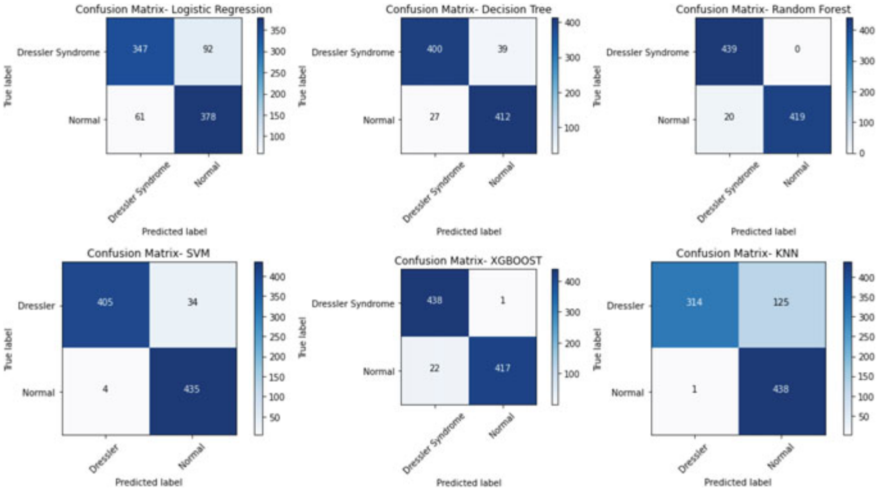


Fig. 7 Confusion matrices with respect to usual implementation

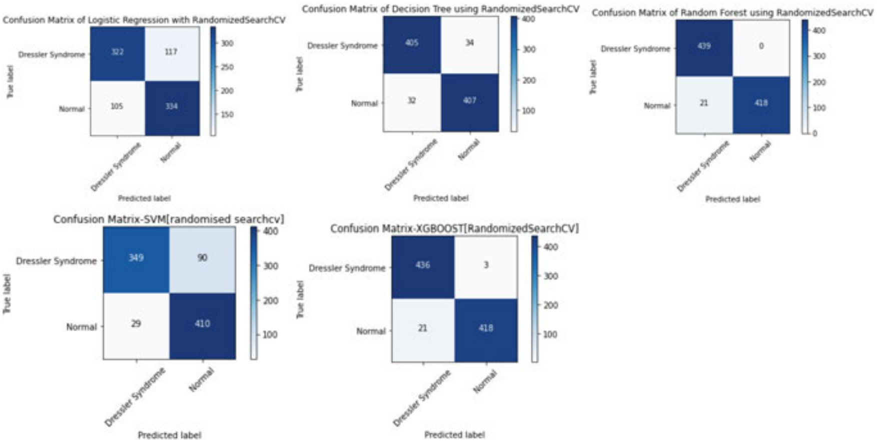


Fig. 8 Confusion matrices with respect to tuned implementation using *RandomisedSearchCV*

From the results analysis it is claimed that since Dressler syndrome is an outcome, hence it is deduced through this study that if symptoms for Dressler syndrome are observed, then it can be helpful in arresting myocardial infarction. The dataset used for this study comprised of 12 labeled values which can be correlated through 111 features.

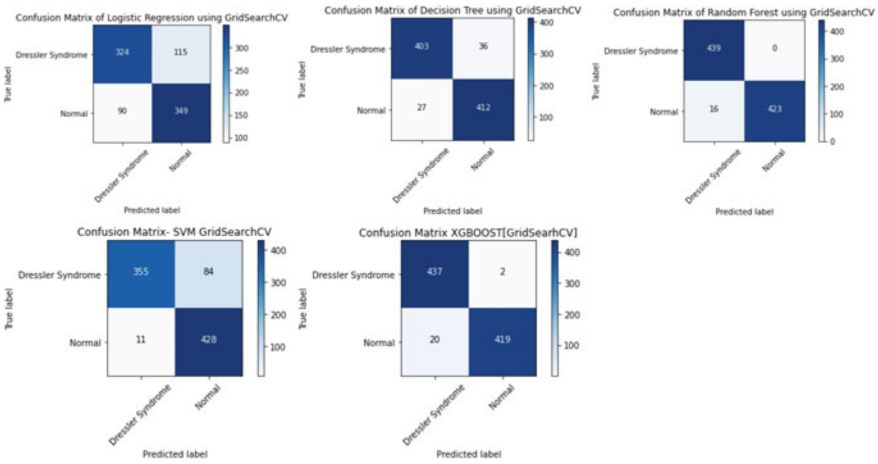


Fig. 9 Confusion matrices with respect to tuned implementation using *GridSearchCV*

5 Conclusion

This study presents the classification of Dressler syndrome using historical myocardial infarction data. In this study, we have used only one label (Dressler syndrome). The other 11 labels can be further classified in the future. Moreover, a uniform classification model can be generated which can classify all the 12 labels accurately using the myocardial infarction data. This study is the first of the twelve classifications based on 12 labels in the dataset.

References

1. Choudhary P, Dandapat S (2020) An evaluation of machine learning classifiers for detection of myocardial infarction using wavelet entropy and eigenspace features. In: 2020 IEEE applied signal processing conference (ASPCON), pp 222–226
2. Fatimah B, Singh P, Singhal A, Pramanick D (2021) Efficient detection of myocardial infarction from single lead ECG signal. Biomed Signal Process Control 68
3. Dressler's syndrome. Cleveland Clinic. https://my.clevelandclinic.org/health/diseases/17947-dresslers-syndrome. Accessed 2 May 2019
4. Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/dresslers-syndrome/symptoms-causes/syc-20371811#:~:text=Dressler%20syndrome%20is%20a%20type,surrounding%20the%20heart%20(pericardium
5. Golovenkin, Shulman, Rossiev DA, Shesternya Myocardial infarction complications Data Set. https://archive.ics.uci.edu/ml/datasets/Myocardial+infarction+complications. Accessed 9 Dec 2020
6. Dohare A, Kumar V, Kumar R (2018) Detection of myocardial infarction in 12 lead ECG using support vector machine. Appl Soft Comput 64(1568–4946):138–147
7. Sun L, Lu Y, Yang K, Li S (2012) ECG analysis using multiple instance learning for myocardial infarction detection. IEEE Trans Biomed Eng 59:3348–3356

8. Ibrahim L, Mesinovic M, Yang K, Eid M (2020) Explainable prediction of acute myocardial infarction using machine learning and Shapley values. IEEE Access 8:210410–210417
9. Degerli A, Zabihi M, Kiranyaz S, Hamid T (2021) Early detection of myocardial infarction in low-quality echocardiography. IEEE Access 9:34442–34453
10. Sharma L, Sunkaria R (2018) Inferior myocardial infarction detection using stationary wavelet transform and machine learning approach
11. Hadanny A, Shouval R, Wu J, Shlomo N (2021) Predicting 30-day mortality after ST elevation myocardial infarction: machine learning-based random forest and its external validation using two independent nationwide datasets. J Cardiol 78(5):439–446
12. Tay D, Poh C, Reeth E, Kitney R (2015) The effect of sample age and prediction resolution on myocardial infarction risk prediction. IEEE J Biomed Health Inform 19(3):1178–1185
13. Kayyum S, Miah J, Shadaab A, lIslam M (2020) Data analysis on myocardial infarction with the help of machine learning algorithms considering distinctive or non-distinctive features. In: 2020 international conference on computer communication and informatics (ICCCI), pp 1–7
14. Omar N, Dey M, Ullah M (2020) Detection of myocardial infarction from ECG signal through combining CNN and Bi-LSTM. In: 2020 11th international conference on electrical and computer engineering (ICECE), pp 395–398
15. Martin H, Izquierdo W, Cabrerizo M, Cabrera A (2021) Near real-time single-beat myocardial infarction detection from single-lead electrocardiogram using long short-term memory neural network. Biomed Signal Process Control 68(1746–8094)
16. Bhaskar N (2015) Performance analysis of support vector machine and neural networks in detection of myocardial infarction. Procedia Comput Sci 46(1877–0509):20–30
17. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python 2825–2830

Analysis of Twitter Data for Business Intelligence

Ishmeet Arora, Apurva Chaudhari, and Sulochana Madachane

Abstract Enhancement of any business requires feedback from customers. This feedback plays a crucial role in knowing the strengths and weaknesses of any business. Gaining these insights these days has become very simple. They are available in the form of—website reviews, social media, etc. The organizations have employees manually analysing this data to figure out the customer sentiments about their products and services, but this process is very time consuming and prone to human error. This cumbersome process of strategic analysis for business intelligence, can be automated. This can be done in two ways rule-based and statistical. There are various automated tools that perform strategic analysis of this data but they are mostly rule-based systems. To address these challenges, we have proposed a system which will automatically analyse customer reviews which takes tweets from twitter as an input and allows the brands to analyse what makes customers happy or frustrated, so that they can tailor products and services to meet their customers' needs. So, in this paper we extract tweets about Samsung mobiles from twitter and use them to analyse which aspects of the product in this case mobiles are performing well and which are not and derive business intelligence from the same.

Keywords Business intelligence · Sentiment analysis · Thematic analysis · Topic modelling · Vectorization

I. Arora · A. Chaudhari (✉) · S. Madachane
Department of Computer Engineering, K.C. College of Engineering and Management Studies and Research, Thane, India
e-mail: apurvakchaudhari@gmail.com

I. Arora
e-mail: ishmeet.k.arora@gmail.com

S. Madachane
e-mail: sulochana.madachane@kccemsr.edu.in

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_7

1 Introduction

Business Intelligence is a data-driven process for making important business decisions. It includes collection, analysis and visualization of data which helps the managers plan important business strategies. It helps them in making informed decisions. These days people are very active on social media and are more vocal than ever before about their opinions on various products, brands, etc. This data is like a treasure trove. Business intelligence can help businesses use this data to adapt to the continuously changing demands of the market. There is a high competition in the industry to retain the customers, companies have the urge to analyse the feedback and evolve over time.

According to the Forrester reports, 74% of firms want to be "data-driven" but only 29% of them are good at connecting analytics to action [1]. From this, we can derive that, businesses need actionable insights to derive business outcomes from data [2].

Many organizations collect feedback from their customers to improve their performance. The organizations need to analyse this feedback to discover insights that would inspire them to drive actions.

Actionable insights are meaningful findings that result from analysing data. They make it clear what actions need to be taken or how one should think about an issue. Organizations use actionable insights to make data-informed decisions [2, p. 1]. Actionable insights can be used to make strategic decisions. These decisions can help derive important outcomes for businesses [2]. It becomes difficult to manually analyse customers' concerns because of the large volume of review data. Hence, our objective is to quickly turn unstructured feedback into insights.

This paper proposed a system to analyse the customer tweets about different organizations, products, services and help the organization to improve their business strategies accordingly. So, our objective is, we will provide a platform where an organizations can analyse the performance of their products and use it to make important business decisions using the reviews by the customer on social media sites.

In this paper we proposed a system which extract real-time tweets about Samsung mobiles from twitter and use them to analyse which aspects of the product in this case mobiles (in general) are performing well and which are not and derive business intelligence from the same.

2 Proposed Method

2.1 Data Extraction

Real-time data, including all recent tweets about Samsung mobiles, is extracted from twitter using the tweepy library.

> Tweepy is an open source Python package that gives you a very convenient way to access the Twitter API with Python [3, p. 1].

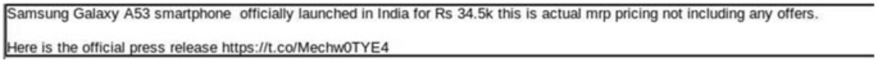
To extract the tweets a query is fired which selects and extracts the tweets according to the keywords used. Some of the keywords used are—Samsung, mobile, service, etc. The query also filters out the tweets containing any kind of media (audios, videos, images). It returns the most recent tweets. For this experiment the number of tweets are limited to merely 1000.

2.2 *Data Preprocessing*

A dataframe is then created for all the extracted tweets. This dataframe contains 2 columns namely tweets and index. Various data cleaning processes are then applied to this dataframe. They include:

- Removing null values—First and foremost all the null values are removed from the dataframe.
- Removing Links—Links would not help in either analysing the sentiment or generating themes. They would only add to the noise. Hence, they are removed (Figs. 1 and 2).
- Removing Punctuations—Some people prefer using proper punctuation in their tweets whereas some people don't. So removing the punctuation would help us treat "amazing!" and "amazing" in the same way.
- Converting emojis to their corresponding text. For e.g.: Happy face smiley (Figs. 3 and 4).

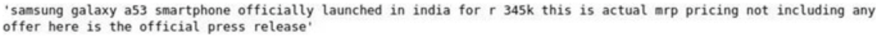
  This would help the proposed system in analysing the sentiments of tweets in the most accurate way possible because most of the time people tend to express their emotions using emojis. An emoji dataset helps in processing the emojis.
- Converting chat words to their corresponding text. For e.g.: lol—laughing out loud.



Samsung Galaxy A53 smartphone officially launched in India for Rs 34.5k this is actual mrp pricing not including any offers.

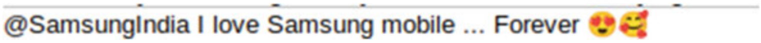
Here is the official press release https://t.co/Mechw0TYE4

Fig. 1 Before removing links



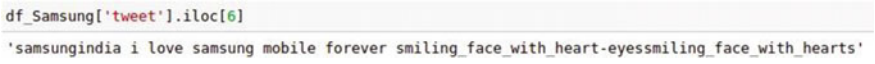
'samsung galaxy a53 smartphone officially launched in india for r 345k this is actual mrp pricing not including any offer here is the official press release'

Fig. 2 After removing links



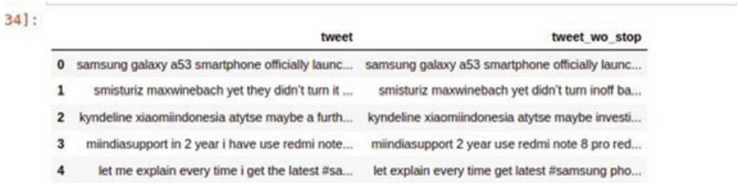
@SamsungIndia I love Samsung mobile ... Forever 😍🥰

Fig. 3 With emojis



```
df_Samsung['tweet'].iloc[6]
'samsungindia i love samsung mobile forever smiling_face_with_heart-eyessmiling_face_with_hearts'
```

Fig. 4 Without emojis



```
34]:
```

	tweet	tweet_wo_stop
0	samsung galaxy a53 smartphone officially launc...	samsung galaxy a53 smartphone officially launc...
1	smisturiz maxwinebach yet they didn't turn it ...	smisturiz maxwinebach yet didn't turn inoff ba...
2	kyndeline xiaomiindonesia atytse maybe a furth...	kyndeline xiaomiindonesia atytse maybe investi...
3	miindiasupport in 2 year i have use redmi note...	miindiasupport 2 year use redmi note 8 pro red...
4	let me explain every time i get the latest #sa...	let explain every time get latest #samsung pho...

Fig. 5 Tweets with and without stop words

These days people have started using abbreviations or chat words very frequently in their tweets or messages. So, converting them into proper text is extremely important for proper semantic analysis of the tweet. We use a dictionary with chat words as keys and their corresponding text as values for this process.

- Removal of Stop Words—Stop words like "a", "the", "is", "are", etc. are removed because they would not be helpful in generating the themes. They would only increase noise. So the stop words are removed and a new column is created in the dataframe which holds all the tweets without stop words (Fig. 5).
- Lemmatization—Lemmatization removes affixes from the word and returns its root form or normalized form [4]. When all the words are in their root form the complexity in analysing is reduced to a great extent, since the basic meaning can be easily deduced from the root words. Hence, we have lemmatized the tweets in the dataframe.
- Tokenization—Tokenization is the process of breaking raw text into words or sentences [5]. We tokenize all the tweets without stop words into a list of words for the purpose of vectorization later on.

2.3 Sentiment Analysis

Sentiment analysis of the tweets is done using the TextBlob library of nltk (Natural Language Toolkit) to predict the sentiment of our tweets in an unsupervised manner. TextBlob is a python library and provides a simple API to perform basic NLP tasks like sentiment analysis, parts of speech tagging, noun phrase extraction, etc. [6]. Using TextBlob we dynamically predict the sentiment for our corpus without having to train a model. This was extremely beneficial as data keeps changing dynamically every time we run the software. Labels used include −1 for negative, 0 for neutral, and 1 for positive tweets. These labels are then stored in the dataframe in the column sentiment across their corresponding tweets.

2.4 Vectorization

A Term Frequency–Inverse Document Frequency (Tf–Idf) Vectorizer is then used to convert string data into numeric form. This is an algorithm used to transform text into a meaningful representation of numbers [7]. It gives weight to each word in every document depending on their importance in the document. A high weight of the Tf–Idf calculation is reached when we have a high term frequency (tf) in the given document and a low document frequency of the term in the whole collection [7]. It considers the overall weightage of a word in the collection of documents. The general assumption is that the word with maximum frequency is important but those could also include words like "this" or "which" which are used very frequently in the English language but don't actually carry any importance. Hence it down weights such words to be able to get the words that are actually important. The vectorizer creates an output Matrix of important TF–IDF features [8].

2.5 Thematic Analysis

Thematic analysis is a method of analysing qualitative data [9, p. 1]. This method examines the data to identify common themes—topics, ideas and patterns that are used repeatedly in the tweets [9]. These themes in our case are basically the topics being discussed the most, among the masses, about Samsung mobiles.

To perform thematic analysis or to identify these topics from the tweets, NMF or Non-Negative Matrix Factorization topic modelling algorithm is used.

According to Chirag Goyle: Non-Negative Matrix Factorization is a statistical method that is used to reduce the dimension of the input corpora [10]. It gives comparatively less weightage to the words that are having less coherence using factor analysis [10]. It works in the following manner:

Input includes the Term-Document Matrix and the number of topics to be generated.

The output gives two non-negative matrices including—words by topics and topics by the original documents.

According to the Fig. 6, the input matrix is decomposed into the following two matrices,

First matrix: It consists of every topic and what words make up that particular topic.

Second matrix: It represents which document includes which topics. Here, linear algebra is used for topic modelling [10].

In our case the number of topics is not fixed. The Gensim library is used to figure out the best number of topics via coherence score. Coherence score is a measure of how interpretable a topic is to humans [11]. According to Enes Zvornicanin:

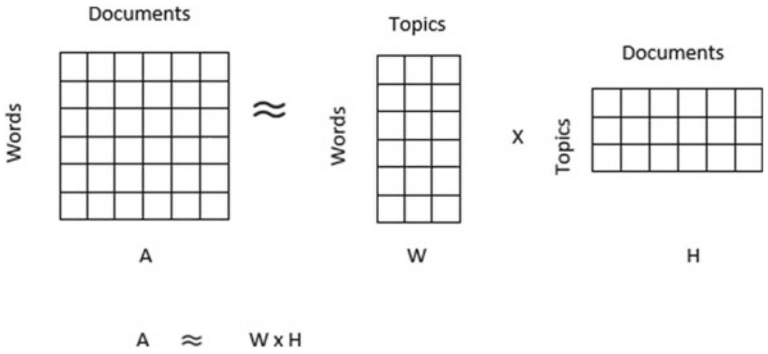


Fig. 6 NMF matrix factorization

> Topics are represented as the top N words with the highest probability of belonging to that particular topic. Briefly, the coherence score measures how similar these words are to each other [11, p. 1].

There is no one way to determine whether the coherence score is good or bad. The score and its value depend on the data that it's calculated from. For instance, in one case, the score of 0.5 might be good enough but in another case not acceptable. The only rule is that we want to maximize this score. Usually, the coherence score will increase with the increase in the number of topics. This increase will become smaller as the number of topics gets higher. The trade-off between the number of top topics and coherence score can be achieved using the so-called elbow technique. The method implies plotting the coherence score as a function of the number of topics. We use the elbow of the curve to select the number of topics.

The idea behind this method is that we want to choose a point after which the diminishing increase of coherence score is no longer worth the additional increase of the number of topics [11, p. 1].

Figure 7 clearly shows that the best number of topics for us is 10.

Initially, a dictionary is created which is basically a mapping between words and their integer id. Then, extremes are filtered out to limit the number of features. Next a list of topic numbers we want to try is created. Next NMF model is run and coherence score is calculated for each number of topics. According to the coherence score best number of topics are selected.

This number is then input with the term document matrix to get the output. The 2 matrices generated in the output tell us which tweet belongs to which topic and what words come under those topics.

Figure 8 shows the words that belong to the 10 selected topics.

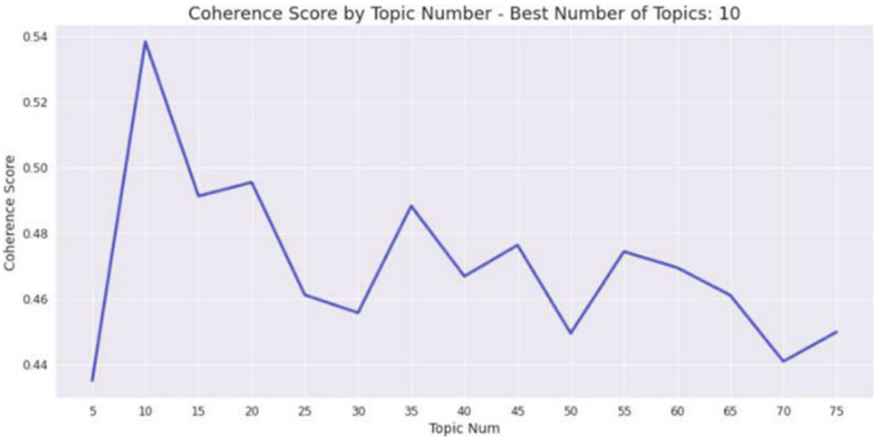
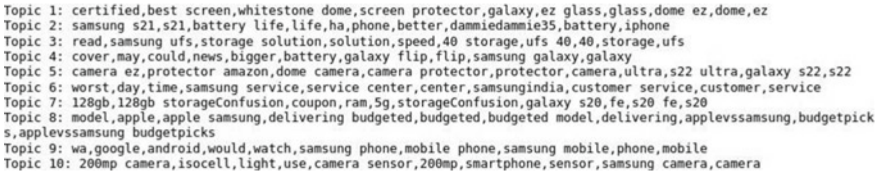


Fig. 7 Coherence score



```
Topic 1: certified,best screen,whitestone dome,screen protector,galaxy,ez glass,glass,dome ez,dome,ez
Topic 2: samsung s21,s21,battery life,life,ha,phone,better,dammiedammie35,battery,iphone
Topic 3: read,samsung ufs,storage solution,solution,speed,40 storage,ufs 40,40,storage,ufs
Topic 4: cover,may,could,news,bigger,battery,galaxy flip,flip,samsung galaxy,galaxy
Topic 5: camera ez,protector amazon,dome camera,camera protector,protector,camera,ultra,s22 ultra,galaxy s22,s22
Topic 6: worst,day,time,samsung service,service center,center,samsungindia,customer service,customer,service
Topic 7: 128gb,128gb storageConfusion,coupon,ram,5g,storageConfusion,galaxy s20,fe,s20 fe,s20
Topic 8: model,apple,apple samsung,delivering budgeted,budgeted,budgeted model,delivering,applevssamsung,budgetpick
s,applevssamsung budgetpicks
Topic 9: wa,google,android,would,watch,samsung phone,mobile phone,samsung mobile,phone,mobile
Topic 10: 200mp camera,isocell,light,use,camera sensor,200mp,smartphone,sensor,samsung camera,camera
```

Fig. 8 Topics with their corresponding words

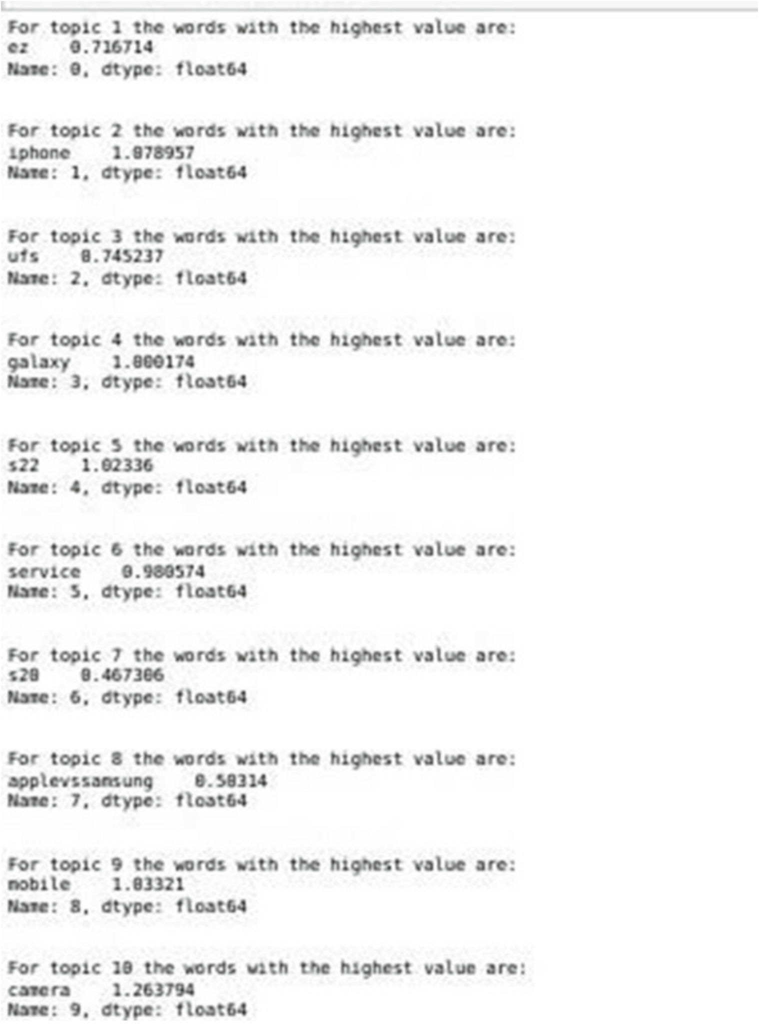
2.6 Feature Extraction

Here By simply iterating in the above two created matrices we figure out how many topics have been generated and which words belong to which topic. Then the tweets are classified according to the topics generated and thus each tweet is assigned a topic. A new column "topic" is created in our dataframe which contains topic numbers across their corresponding tweets. Now the number of positive, negative and neutral tweets for every topic is calculated and a separate dataframe is created for the same. Also, total number of positive, negative and neutral tweets is calculated. Various graphs using these values are plotted and displayed (Fig. 9).

3 Results

Using the above-explained method and dataframe created, we can easily generate graphs and draw business intelligence from them.

Figure 10 is a pie chart that represents the total no of positive, negative and neutral tweets among all the tweets extracted. It is clear from the figure that the number of



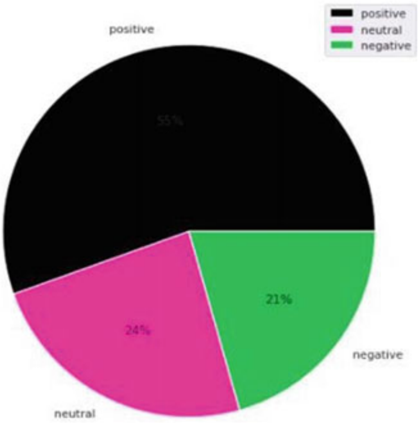
```
For topic 1 the words with the highest value are:
ez    0.716714
Name: 0, dtype: float64


For topic 2 the words with the highest value are:
iphone    1.078957
Name: 1, dtype: float64


For topic 3 the words with the highest value are:
ufs    0.745237
Name: 2, dtype: float64


For topic 4 the words with the highest value are:
galaxy    1.000174
Name: 3, dtype: float64


For topic 5 the words with the highest value are:
s22    1.02336
Name: 4, dtype: float64


For topic 6 the words with the highest value are:
service    0.980574
Name: 5, dtype: float64


For topic 7 the words with the highest value are:
s20    0.467306
Name: 6, dtype: float64


For topic 8 the words with the highest value are:
applevssamsung    0.58314
Name: 7, dtype: float64


For topic 9 the words with the highest value are:
mobile    1.03321
Name: 8, dtype: float64


For topic 10 the words with the highest value are:
camera    1.263794
Name: 9, dtype: float64
```

Fig. 9 Words with the highest value for every topic

positive tweets is more than negative or neutral tweets. This observation can be used to derive the inference that the overall customer sentiment about Samsung mobiles is positive.

Figure 11 is a bar graph of topics vs the number of tweets and the sentiment of those tweets. We have used three colours yellow—depicting positive sentiment, purple—depicting neutral sentiment and blue—depicting negative sentiment. For every topic we can see the number of tweets that belong to that particular topic and also the sentiment of those tweets. We can clearly see topic 2 has the maximum number of tweets. This tells us that topic 2 is the most popular topic among the customers.

Fig. 10 Sentiment analysis



From Figs. 8 and 9 we can see that topic 2 is about battery and iPhone. The colour scheme used in the bar is a mix of all three colours, no colour is dominant, which specifies neutral emotions. We can also see that the topic with max positive sentiment is topic 1 which represents screen, screen protector, glass—basically hardware. This shows that the customers are happy with the hardware. Also, it is clear that topic 5 is performing badly which is evident from the fact that the most dominant colour in the bar is blue. From Fig. 9. we observe that topic 6 represents the feature "service" or "customer service". Hence, we can conclude that customers are not happy with the customer service and it needs more work.

Figure 12 is a dataframe we created. It consists of 4 columns which include topic names and the total number of positive, negative and neutral tweets about that particular topic. This dataframe can be used to create various different graphs and hence analyse the data in various different ways.

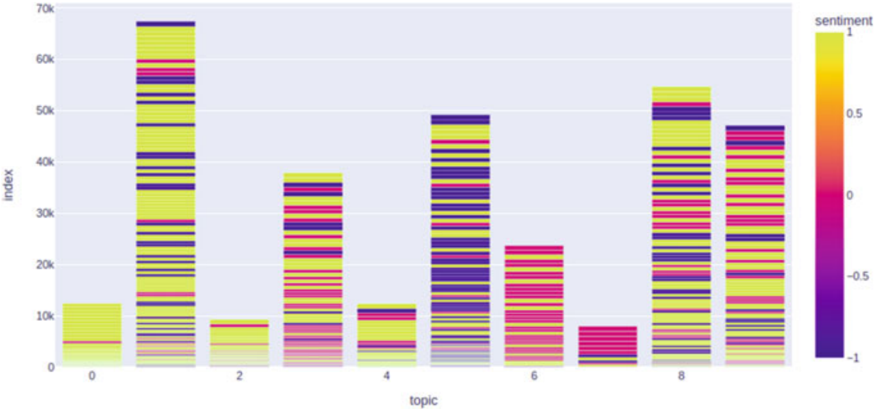
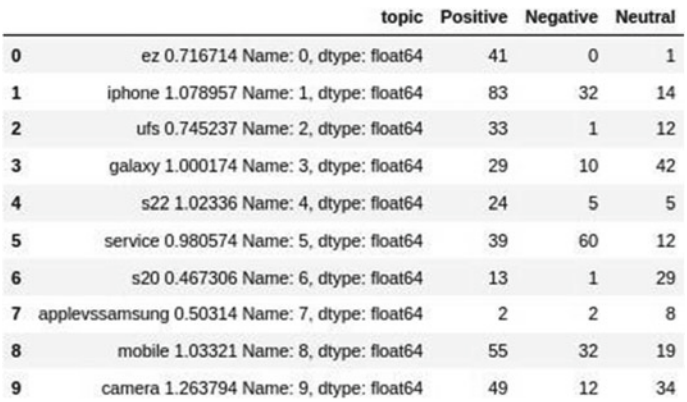


Fig. 11 Topics versus number of tweets and their sentiments



	topic	Positive	Negative	Neutral
0	ez 0.716714 Name: 0, dtype: float64	41	0	1
1	iphone 1.078957 Name: 1, dtype: float64	83	32	14
2	ufs 0.745237 Name: 2, dtype: float64	33	1	12
3	galaxy 1.000174 Name: 3, dtype: float64	29	10	42
4	s22 1.02336 Name: 4, dtype: float64	24	5	5
5	service 0.980574 Name: 5, dtype: float64	39	60	12
6	s20 0.467306 Name: 6, dtype: float64	13	1	29
7	applevssamsung 0.50314 Name: 7, dtype: float64	2	2	8
8	mobile 1.03321 Name: 8, dtype: float64	55	32	19
9	camera 1.263794 Name: 9, dtype: float64	49	12	34

Fig. 12 Topic and sentiment

4 Discussion

The extraction of relevant real-time data from twitter is one big challenge. This is because initially the extracted data is filled with noise. Most of the tweets include promotional tweets, media like audios, videos and images, links to youtube videos. Such data constitutes 65% of the tweets extracted. This challenge can be easily overcome by adding specific keywords and filters to the query used. Keywords that we use include:

"Samsung", "mobile", "service" and many more.

The tweets are then pre-processed and their sentiments analysed. Next they are vectorized. Vectorization can be done in two ways—(1) by using a Count Vectorizer and (2) by using a Tf–Idf Vectorizer. We use Tf–Idf vectorizer. The reason for this is that:

Count Vectorizer only counts the frequency of the appearance of a word in the document which results in biasing in the favour of most frequent words. Due to this, rare words are ignored which could have helped in processing our data more efficiently [12]. To overcome this, we use Tf–Idf Vectorizer. Tf–Idf Vectorizer considers overall document weightage of a word [12]. It downweights those words which occur frequently but do not have any significant importance to the context of the sentence [12].

Tf–Idf Vectorizer assigns a greater weight to those words which are less frequent or rare [12]. It considers the occurrence of a word in the entire corpus instead of considering its occurrence in a single document [12].

The Tf–Idf vectorizer creates a term-document matrix which is fed into the NMF topic modelling algorithm. Along with this matrix, the number of topics or themes are also needed as input to the algorithm. These number of topics should be decided on the basis of the kind of data one is dealing with. Since we have little idea about the kind of tweets extracted, and they change for every execution, it is best to keep the number of topics dynamic to maintain the accuracy of the results. This is where the coherence score comes into the picture. According to Enes Zvornicanin:

We can use the coherence score in topic modelling to measure how interpretable the topics are to humans. In this case, topics are represented as the top N words with the highest probability of belonging to that particular topic [11, p. 1].

The coherence metrics used is called CV. It creates content vectors of words using their co-occurrences and then calculates the score using normalized pointwise mutual information (NPMI) and the cosine similarity [11]. This metric is the default metric in the Gensim topic coherence pipeline module [11]. This measure does have some drawbacks though. After many trials and tests Michael Roeder, Member of Data Science Group at UPB, has come to the conclusion that "it behaves not very good when it is used for randomly generated word sets [13, p. 1]". But in our case we are not randomly generating our tweets so it works well.

For the purpose of generating themes we use a topic modelling algorithm. There are various topic modelling algorithms but we use NMF. They include Latent Semantic Analysis (LSA), Non-Negative Matrix Factorization (NMF), Latent Dirichlet Allocation (LDA), Parallel Latent Dirichlet Allocation (PLDA) and Pachinko Allocation Model (PAM). LSA focuses more on matrix dimension reduction whereas LDA and NMF focus on solving topic modelling problems [14]. So this rules LSA out. LDA works better on a corpus containing large documents whereas NMF works better on a corpus containing smaller documents [15]. A document in our case represents a single tweet hence NMF is a better choice. Also, in a study about comparison between LDA and NMF it was observed that the execution time of NMF is lower than the execution time of LDA [16]. It also observed that NMF secured a better coherence score as compared to LDA [16]. PLDA and PAM are improvised versions of LDA. Hence NMF is the best choice for topic modelling.

There is one disadvantage though, the time complexity of NMF topic modelling is polynomial [17]. It is an NP-hard problem, which means it is difficult to find an optimal solution [18]. This problem can be solved using Hierarchical Alternating Least Squares Algorithm for NMF (HALS–NMF) [18]. Another common practice to approach NP-hard problems is to use gradient descent [18].

5 Conclusion

BI has become essential to all sizes of organizations as everything has become digital and people are more aware about their surroundings and the variety of options present. The competition in the market is ever-increasing. In today's world, to sustain in this market a company has to implement BI. BI is expected to grow exponentially in the future.

We have proposed a method using which a platform (interface) can be created, where any organization can not only see customer reviews about their products but can also use, the analysis done and represented in a graphical format, to make important business strategies and improve their performance.

The method includes performing sentiment and thematic analysis on the reviews and extracting features from the themes generated. The organizations can use this platform to see which features are performing badly, why and what areas need more work. For example, from Fig. 9, it is clear that topic 6 is performing badly which is evident from the fact that among all the tweets about it, maximum tweets have a negative sentiment (blue colour). From Fig. 9, we observe that topic 6 represents the feature "service" or "customer service". Hence, we can conclude that customers are not happy with the customer service and it needs more work.

Similarly, many different kinds of graphs can be created and different kinds of analysis can be done which will help the organizations understand customers' needs and make changes accordingly.

Thus, business intelligence can be of great help to organizations and help facilitate their growth.

References

1. Hopkins B (2017) Think you want to be 'data-driven'? Insight is the new data. Forrester. www.forrester.com/blogs/16-03-09-think_you_want_to_be_data_driven_insight_is_the_new_data. Accessed 8 May 2022
2. Medelyan A (2021) 3 examples of actionable insights from customer feedback analysis. Thematic. www.getthematic.com/insights/how-to-get-actionable-insights-from-your-customer-feedback-analysis. Accessed 8 May 2022
3. Real Python (2021) How to make a Twitter Bot in Python with Tweepy. www.realpython.com/twitter-bot-python-tweepy. Accessed 8 May 2022
4. Sawhney P (2022) Introduction to stemming and lemmatization (NLP)—Geek Culture. Medium. www.medium.com/geekculture/introduction-to-stemming-and-lemmatization-nlp-3b7617d84e65. Accessed 8 May 2022
5. Chakravarthy S (2021) Tokenization for natural language processing—Toward Data Science. Medium. www.towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4. Accessed 8 May 2022
6. TextBlob: simplified text processing—TextBlob 0.16.0 documentation. Steven Loria. www.textblob.readthedocs.io/en/dev. Accessed 9 May 2022
7. Chaudhary M (2021) TF-IDF Vectorizer Scikit-Learn—Mukesh Chaudhary. Medium. www.medium.com/mukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a. Accessed 9 May 2022
8. Sklearn.Feature_extraction.Text.TfidfVectorizer. Scikit-Learn. www.Scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. Accessed 9 May 2022
9. Caulfield J (2022) How to do thematic analysis | A step-by-step guide and examples. Scribbr. www.scribbr.com/methodology/thematic-analysis. Accessed 9 May 2022
10. Goyal C (2021) Topic modelling using NMF | Guide to master NLP (part 14). Analytics Vidhya. www.analyticsvidhya.com/blog/2021/06/part-15-step-by-step-guide-to-master-nlp-topic-modelling-using-nmf. Accessed 9 May 2022
11. Zvornicanin E (2021) When coherence score is good or bad in topic modeling? Baeldung on Computer Science. www.baeldung.com/cs/topic-modeling-coherence-score. Accessed 9 May 2022
12. Goyal C (2021) Text vectorization and word embedding | Guide to master NLP (part 5). Analytics Vidhya. www.analyticsvidhya.com/blog/2021/06/part-5-step-by-step-guide-to-master-nlp-text-vectorization-approaches. Accessed 9 May 2022

13. Roeder M Not being able to replicate coherence scores from paper issue #13 dice-group/Palmetto. GitHub. www.github.com/dice-group/Palmetto/issues/13. Accessed 9 May 2022
14. Ma E (2018) 2 latent methods for dimension reduction and topic modeling. Medium. www.towardsdatascience.com/2-latent-methods-for-dimension-reduction-and-topic-modeling-20ff6d7d547. Accessed 9 May 2022
15. Mifrah S, Benlahmar EH (2020) Topic modeling coherence: a comparative study between LDA and NMF models using COVID'19 corpus. Int J Adv Trends Comput Sci Eng. https://doi.org/10.30534/ijatcse/2020/231942020
16. George S, Vasudevan S (2021) Comparison of LDA and NMF topic modeling techniques for restaurant reviews
17. Topic extraction with non-negative matrix factorization and latent Dirichlet allocation. Scikit-Learn. www.scikitlearn.org/stable/auto_examples/applications/plot_topics_extraction_with_nmf_lda.html. Accessed 9 May 2022
18. An J (2020) Nonnegative matrix factorization problem. Undergraduate Honors Theses, Paper 1518. https://scholarworks.wm.edu/honorstheses/1518

Detection and Classification of Cyber Threats in Tweets Toward Prevention

Sayanta Harh, Sourav Mandal, and Debasis Giri

Abstract The Internet has become a vital aspect of everyone's life in the twenty-first century. As the number of people using the Internet grows, so is the number of cyberattacks. Over the years, extensive research has been conducted to detect cyber threats from several online sources. This work was also done with this goal in mind. We picked Twitter as the information source and attempted to order a tweet to fall into digital danger classification or not, further arranging it into different subcategories like, DDOS, Ransomware, Malware, and so on. We used bidirectional long short-term memory (BiLSTM) as a recurrent neural network (RNN) augmentation on two levels (multilevel classification). At the most basic level, we used BiLSTM to divide tweets into four categories, one of which is that they pose a cyber threat, which we classified as a threat. At the next level, we classified the threat categories into seven subcategories of threat types. In level-1 classification, we outperformed similar systems with a test accuracy of 88.16% on the whole dataset and 88.08% accuracy on test dataset with 30% split, while in level-2 classification of threat tweets (followed by level-1) into its subcategories, we obtained a test accuracy of 81.71%.

Keywords Cyber threat identification · Common vulnerabilities and exposure · Cyber threats classification · Bidirectional long short-term memory (BiLSTM)

1 Introduction

The detection and classification of cyber risks from a stream of data, such as tweets or a string of text, are a piece of work we've completed effectively. The standard dataset we used manually annotates tweets in the four categories of 'Irrelevant,' 'Marketing,' 'Threat,' or 'Unknown.' The 'Threat' tweets are then further classified

S. Harh · D. Giri
Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Kolkata, West Bengal, India

S. Mandal (✉)
School of Computer Science and Engineering, XIM University, Bhubaneswar, Odisha, India
e-mail: sourav.mandal@ieee.org

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_8

into seven subcategories (as available in the dataset)—'vulnerability,' 'ransomware,' 'Ddos,' 'leak,' 'general,' '0day,' and 'botnet.' The 'Threat' category is for tweets that contain cyber threat clues like words or phrases, such as 'I will hack the "XYZ" bank tomorrow.' This statement clearly contains some cyber threat information. The 'irrelevant' tag is used to tweets that do not contain any information on cyber dangers, such as 'The sun rises from the east.' This comment has no bearing on how cyber risks are classified. 'Would you want to get the subscription of antivirus "ABC" to defend yourself from ransomware attacks at a 50% discount?' is an example of a tweet having the 'Marketing' tag applied to it. The phrases ransomware and antivirus appear in this tweet, although they have no bearing on cyber dangers. Finally, the 'Unknown' category includes tweets that contain cyber threat taxonomy but are uncertain whether they contain cyber threat relevant material, such as 'DDoS attack tutorial @ http://y.tube/57rTuiOv.' This tweet can be utilized by people with both positive and negative mentalities. In terms of subcategorization, the category 'Vulnerability' refers to any information that reveals a software or hardware vulnerability; for example, 'Windows 8.1 service pack 1 has a security update that renders it vulnerable to remote access.' When ransomware attack information is provided out, the type 'Ransomware' is annotated, for example, 'Company ABC suffered a significant ransomware attack with 256-bit encryption.' The remaining classes are similarly labeled with their literal definitions. To achieve a better comparison with [1], we proposed a new BiLSTM-based classifier and tested it using [1]'s provided dataset, which contains manually labeled tweets. Table 1 illustrates several samples of tweets and how they were classified according to the cyber threat taxonomy.

As the world's population grows, so does the number of people who use the internet, and cyberattacks are becoming more regular. The fundamental goal that

Table 1 General classification of tweets involving cyber threats

Tweets/text	Keywords	Category (class)	Subcategory
Pokémon go crashed due to a massive DDoS attack from an unknown source	DDoS	Threat	DDoS
Sigmoid is a new ransomware malware launched by Anonymous crew, which is able to encrypt devices with 256-bit encryption	Sigmoid, ransomware, malware, encrypt, 256-bit, encryption	Threat	Ransomware
Nord VPN 6.14.31 Denial of Service available at 50% discounted rate: https://t.co/ZdIzHsDY4b	Denial of Service, VPN, discounted rate	Marketing	Other
Hack the box is a great website for bug-bounty	Hack	Irrelevant	Other
Hackersploit is a YouTube channel that posts regular hacking-related videos	Hackersploit, hacking	Unknown	Other

drove us was to try to prevent cybercrime from occurring in the future. This drive inspired us to construct this model, which is currently a work in progress. The first issue we encountered when beginning the investigation was obtaining an appropriate dataset. Behzadan et al. [1] attempted a similar problem and published this dataset for the first time. They gathered the data, labeled them, and uploaded the dataset to their GitHub[1] profile using the TWINT API.[2] We used their dataset to complete all the tasks for our proposed system, then compared the results to [1]. They gathered tweets about cyber dangers using a TWINT API filter. They employed a cybersecurity taxonomy as a filter, which contains terms like 'ransomware,' 'DDoS,' and 'hacking,' among others. The next step was to locate appropriate neural networks and tune them to improve accuracy. Finally, we go with BiLSTM [2] network, modify it accordingly to propose a better model.

Now, open-source intelligence (OSINT)[3] is a great source of information about newly discovered scam techniques and trending vulnerabilities, which, combined with the database of national vulnerability database (NVD),[4] aids researchers and ethical hackers in developing better solutions to prevent hacking. Similarly, the common vulnerability and exposure (CVE) database[3] contains detailed information about any newly discovered threat, allowing defensive security researchers to develop countermeasures to prevent vulnerabilities from being exploited again. After receiving the dataset, it is converted to an appropriate format so that it can be sent to the sequential RNN-based BiLSTM neural network. The result of our model then shows the likelihood of the input tweet falling into the following categories: 'Threat,' 'Irrelevant,' 'Marketing,' or 'Unknown.' Let's say we have a tweet of 'Unknown' type that says, 'Facebook Patched a Remote Code Execution Vulnerability.' This tweet will produce the following outcomes with the probabilities: Threat level: 0.1, Irrelevant level: 0.01, Marketing level: 0.1, and Unknown level: 0.79. Let us again consider another example of 'Threat' category, 'Pokémon go underwent massive DDoS attack after the successful launch event,' this would give the output in the following order: Threat level: 0.95, Irrelevant level: 0.01, Marketing level: 0.1, and Unknown level: 0.3. The outcome with the highest probabilistic value then is classified as final, and the desired output is eventually attained. This is the general working principle of any standard classification algorithm to output final class (like using argmax function for Naïve Bayes algorithm). We employed classifiers on two levels, with two comparable BiLSTM-based neural networks ensemble in multilevels, to classify the tweets. The level-1 (coarse-grained) classification of the input tweets is detailed in the preceding paragraph. To the next level, we split the 'threat' category tweets and used them as a separate dataset. Following level-1 classification, the 'Threat' tweets were given to the level-2 classifier, which was developed using the same process as the multi-class classifier used in the first-level classifier (coarse-grained). The threat comprising tweets was then classified into seven subcategories—'vulnerability,' 'ransomware,'

[1] https://github.com/behzadanksu/cybertweets.

[2] https://github.com/twintproject/twint.

[3] https://osintframework.com/.

[4] https://nvd.nist.gov/.

‘Ddos,’ ‘leak,’ ‘General,’ ‘0day,’ and ‘botnet.’ This stage is like the one described in the previous paragraph, in that the likelihood is computed and the most likely outcome is chosen as the final class. For example, a tweet stating that ‘Facebook had the largest data breach of the millennium’ would result in the following output: 0.01, ransomware: 0.02, DDoS: 0.01, leak: 0.94, General: 0.01, zero-day: 0.005, botnet: 0.005. Again, the highest probable outcome—‘leak’ is selected automatically as final class by the SoftMax layer of the neural architecture (see Figs. 2 and 3 of Sect. 3). Some of the challenges raised in this study are listed below.

- The most challenging hurdle was acquiring a standard dataset; we had a lot of issues because this type of dataset was not publicly available, thus we had to rely on a dataset from [1].
- We had to figure out how to solve the problem utilizing the best available deep neural network algorithms with a variety of hyper-parameters, as well as hidden layer and dense layer combinations, because various commonly used methodologies for threat classification were already in use.
- We ran into challenges with system resource restrictions while using hyper-parameter tuning to boost classification accuracy.

Some of our specific contributions to this work are listed below.

- Making the multi-class classification of tweets indicated above (see Fig. 1) in multilevel.
- In the proposed BiLSTM neural network, we used alternate combinations of hidden layers and hyper-parameter tuning, which allowed us to achieve a higher degree of accuracy.
- While there are various works in this domain that may classify threats, we designed our model to classify an endless number of threat categories with a single point of modification.

The section after that describes various similar works, followed by our proposed methodology in Sect. 3 with system workflow. In Sect. 4, we discussed the dataset and our system’s performance, followed by a conclusion and future scope in Sect. 5.

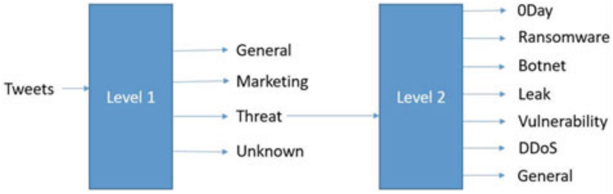


Fig. 1 The multilevel classification of the proposed system

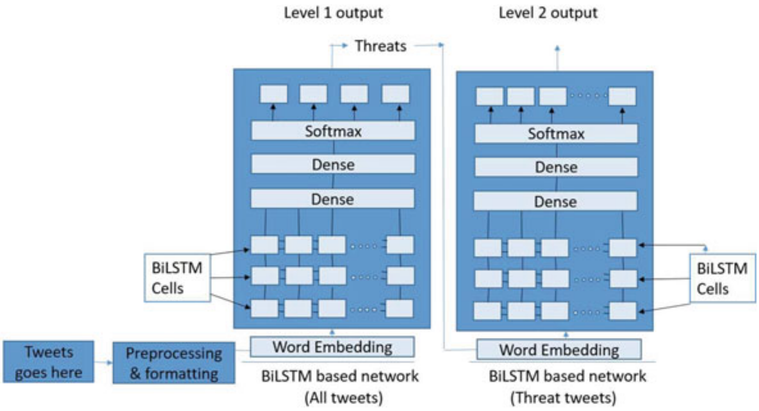


Fig. 2 The workflow of the proposed system model

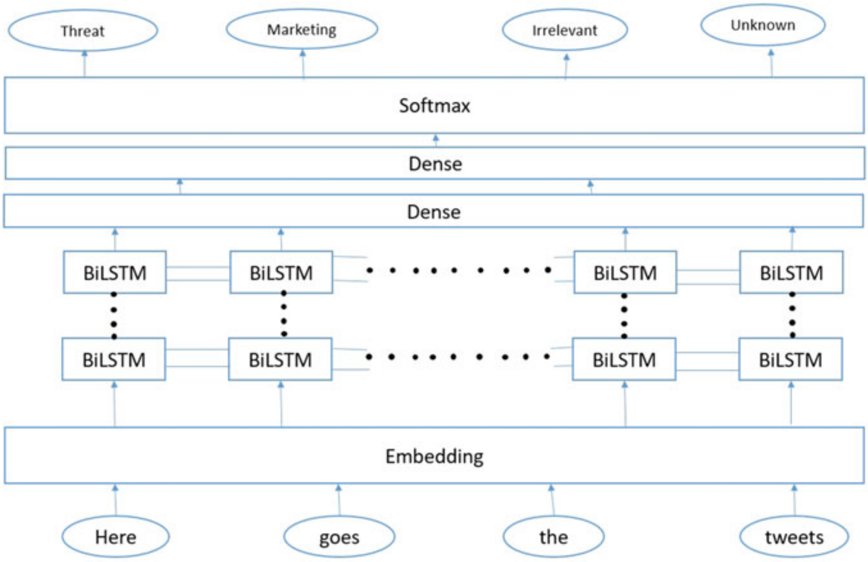


Fig. 3 The BiLSTM-based neural network for level-1 classification

2 Related Work

In the study, Behzadan et al. [1] proposed a method for collecting tweets. They also put together a set of annotated datasets with 21,000 tweets. We used the same dataset for our work. Furthermore, they used Convolutional Neural Network to create a deep neural network that binary analyses tweets before classifying them into numerous threat subcategory classifications. The model work's output receives an F1—score of 0.82, which is comparable to 82%. Bose et al. [3] focused on categorizing new occurrences that occur in the context of a cyber threat as a novel or developing. In several cases, they employed an unsupervised learning algorithm, which is rather uncommon. The ranking system developed as a result of this work is superior to any previously developed system. Their algorithm also includes a technique for ranking trending events as soon as they occur, ensuring that events that are innovative but not hot or widely tweeted do not lose their significance. After constructing the classifier, thirty manually annotated tweets were used to evaluate it, yielding an accuracy of 75% True Positive, 83.33% True Negative, and a precision of 93.75%. Sceller et al. [4] presented an automated approach that can detect and classify cyber threat-containing tools in real time in their study. Sonar is the name of the program they created. It's a graphical tool that can provide real-time classification notifications as well as the geological location from which the tweets are coming. They fed their nearest neighbor algorithm with 47.8 million tweets. Furthermore, their technology is capable of comprehending not only English, but every language is spoken anywhere on the planet. Because the time complexity is O(c), or constant time, Sonar can provide real-time notifications. In the study, Le et al. [5] established the concept of threat intelligence collecting using tweets. They improved the novelty categorization by using a neural network. The primary purpose was to collect all cyber threat intelligence (CTI) and format it in a specific way. They adopted the same formatting as the CVE database, and they worked to ensure that the CVE is updated faster than it is now. They had a 64.30% accuracy rate. Their work was also compared to that of other researchers employing well-known approaches such as conventional SVM, CNN, and multi-layered perceptron (MLP), and it was discovered that their work provided greater precision and F1-score. In the research, Dionísio et al. [6] created a model that can binary classify tweets and then utilize named entity recognition to further classify them into their appropriate classification. They've developed a deep neural network-based processing pipeline for a revolutionary tool. After separating the dataset into three parts, they collected data for four months and used it for training, testing, and validation. They employed CNN and produced a True Positive rate of 94% and a True Negative rate of 91%, which is much higher than many other models. Their model also gets a 92% F1-score for the NER service it delivers. We discovered that the accuracy obtained by the systems described above is less. Many researchers have sought to improve threat classification, but we have proposed a system that allows us to classify threats over a wide range of categories, compared to currently available systems. In comparison to the previous research, we attained substantially higher classification accuracy.

3 Proposed Methodology and System Workflow

3.1 Data Collection

We largely used a standard dataset compiled by [1]. As previously said, anyone who uses Twitter is sitting on a gold mine waiting to be discovered. The Twitter Intelligence API, abbreviated as 'TWINT,' was utilized. TWINT is a python-based library that may be easily imported, and installation instructions can be found online. Behzadan et al. [1] used TWINT's filters to only use tweets from firms classed as DDOS, Ransomware, Botnet, Vulnerability, Leak, and so on.

The following subsections detail the proposed system's design. Figure 1 depicts the primary workflow of our proposed approach. The system components are shown in greater detail in Fig. 2. The BiLSTM cells (small rectangle blocks), which are effectively two LSTM employed one for forward computation and the other for backward computation, are depicted in Fig. 2. In addition, the proposed system is based on the work of [7].

3.2 Implementation

As previously stated, we employed BiLSTM in this project. We started by importing the data. Because the data was derived from tweets, some preprocessing was required, such as the removal of punctuation, special characters, and mentions, among other things [8–10]. After that, lemmatization [11] and tokenization [12] were used. As we all know, lemmatization scrapes a word by its postfix, thus related terms like 'simple', 'simpler', and 'simplest' all have the same weight, which could have resulted in an error in the weight computation of individual words. Tokenization is another useful step since it splits strings into meaningful tokens, which helps the computer calculate the weight of the words as a sequence more accurately use for further vectorization. Following that, the data was sent to the neural network model via the embedding layer, which created the embedding matrix using the glove model [3]. The two dense layers are then followed by stacked BiLSTM layers utilized for computation. The activation function was employed by the SoftMax layer on top to determine the final output class. Over 30 epochs, we iterated the training process and were able to obtain high accuracy.

The split dataset is used to create the second classification model in level-2, which only includes the 'Threat' categories from the level-1 classifiers. We found 9341 tweets that were rated as 'threat' level-1 on Twitter (actual 8351 threat tweets are available in dataset). They were used to train our level-2 classification model, which has a similar fundamental structure to the first (see Fig. 2). As a result, we were able to achieve the best possible training accuracy.

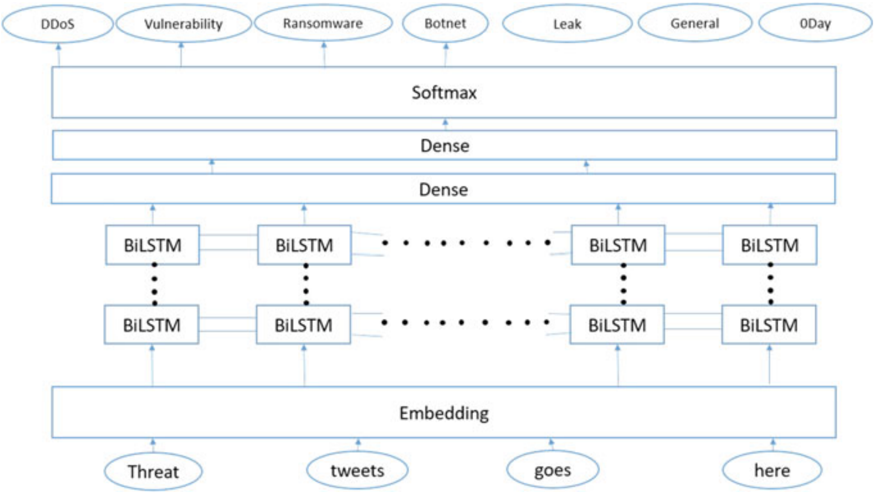


Fig. 4 The layered structure of the proposed classifier with the BiLSTM

Algorithms for cyber threat classification from Twitter using BiLSTM-based deep neural network are given below for both the levels. Figures 3 and 4 also show the deep neural network architecture for the same.

Algorithm 1 The multi-class tweet classification Level-1 Classification (coarse-grained).

Input: Tweets.
Output: Multi-class labeling of tweets.
Step 1: Input tweets, *t*.
Step 2: Pre-process and format *t*.
Step 3: Vectorize the tweets.
Step 4: Learning the model using BiLSTM-based neural network as per Fig. 3.
Step 5: Determine the final class (output) label (threat, marketing, irrelevant, or unknown).
(SoftMax determines the maximum probable outcome as final class)

Algorithm 2 The Level-2 Classification (fine-grained) for threat subcategorization.

Input: Threat-containing Tweet Stream.
Output: Threat subcategorization of tweets.
Step 1: Save the captured tweets, *t*.
Step 2: Pre-process *t*.
Vectorize the tweets and add padding to them.
Step 3: Train the deep learning model using the tweets as per Fig. 4.
Step 4: Determine the final class (output) label.
(SoftMax determines the maximum probable outcome as final class)

Overview of the proposed BiLSTM-based neural network. Next, we describe the different components and layers of our BiLSTM-based neural network. The BiLSTM architecture [13–15] is made up of memory blocks or LSTM cells, which are recurrently connected network modules. A standard BiLSTM classifier computes the hidden vector sequence $h = h_1, h_2 \ldots, hT$ and the output class Y given an input sequence $x = x_1, x_2, \ldots, xT$. The equations that make up the model are as follows.

$$\underset{h_i}{\rightarrow} = \mathrm{LSTM}_{\mathrm{fw}}\left(\underset{h_{i-1}}{\rightarrow}, x_i\right) \tag{1}$$

$$\underset{h_i}{\leftarrow} = \mathrm{LSTM}_{\mathrm{bw}}\left(\underset{h_{i+1}}{\leftarrow}, x_i\right) \tag{2}$$

$$\delta = \mathrm{drop}\left(\underset{h_i}{\rightarrow}, \underset{h_i}{\leftarrow}\right) \tag{3}$$

$$y = \mathrm{softmax}\left(\left[\underset{h_i}{\rightarrow}, \underset{h_i}{\leftarrow}\right]\right) \tag{4}$$

For the classification job, the T + 100-layer model is utilized, which is based on the state-of-the-art BiLSTM presented in [15]. Finally, as a Softmax layer, we added a multi-class classifier to classify the operation. Figure 3 shows the level-1 (coarse-grained) classification for all the tweets.

Figure 4 shows the level-2 (fine-grained) classification for the tweets belong to the threat category.

Embedding layer. The initial layer of our neural network architecture is this layer [9, 11, 16]. The primary goal of this layer is to convert the words into fixed-size vectors. First, using pre-trained word vectors, the word problem is turned into a vector [17, 18]. To produce the sequence of input from the word problem, the widely utilized Glove word embedding [16] is employed. The embedding layer vectorizes, as the name implies. The embedding layer employs the embedding matrix, a large chunk of data that has been successfully generated and dubbed the Glove model. This is one of the most effective vectorization techniques. The following parameters are used: the *input dimension* is the size of individual words, the *output dimension* is 100, the *input length* is the sequence length of words in a sentence, the *weights* are [embedding matrix], and *Trainable* is False.

BiLSTM layer. As previously stated, the layer is the heart and soul of our neural network, performing all of the heavy liftings. BiLSTM (Bidirectional Long Short-Term Memory) [13–15] is an advancement over standard Recurrent Neural Networks (RNN). BiLSTM is a model for processing sequential data available that is among the best. Backpropagation is one of the two methods for minimizing error or optimizing loss. This is how the neural network best understands the sequences and aids in further classification. Even when the amount of data accessible is limited, BiLSTM makes efficient use of it by traversing it back and forth. The following parameters were used: *Return sequence* = True, *weight* = 10.

Dense Layer. The dense layer is a deep-connected neural network layer [19], meaning that each neuron in it receives input from across all neurons in the previous layer. The dense layer produces an 'm' dimensional vector as its output. As a result, the dense layer is mostly employed to alter the vector's dimensions. Some other essential hyper-parameters used in the proposed system are given below.

1. hidden units are 16 and activation function is 'ReLu,' ReLu works with the formula: $G(z) = \max\{0, z\}$
2. hidden units are 64 and activation function is 'ReLu,' ReLu works with the formula: $G(z) = \max\{0, z\}$
3. hidden units = 4 and activation function = 'softmax,' Softmax works with the formula: $\sigma(\underset{Z}{\rightarrow})_{i} = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}}$

   Where, 'σ' represents Softmax output, 'e' represents exponential, 'z' represents token count, 'j' represents iteration variable, 'K' represents learning rate, and finally 't' represents bias.

Optimizer. We used Adam optimizer [20, 21] for our neural network except for the layers. It optimizes very efficiently and is based on the stochastic gradient descent (SGD) principle [22]. When we received noisy data, we utilized Adam optimizer to optimize the neural network model. It automatically modifies network weights and biases, making the model more efficient. During the training of the neural network model, we additionally implemented an early halting mechanism based on parameter value loss [23] to avoid gaining poorer accuracy. This works beautifully and has a track record of success.

Loss function. The loss function of our neural network was also controlled using 'categorical_crossentropy' [8]. In categorical cross-entropy, the loss function is determined as the difference between the expected probability and the actual classes. After putting it to use, we were able to surpass the 80% threshold. Figure 5 shows the training loss and accuracy of our proposed level-1 annotator. We were able to achieve the model's training accuracy of 81.53% and use it to achieve further heights in the following categorization. Figures 5, 6, and 7 show the training and validation accuracy and loss over the no of epochs for our level-1, level-2, and level-2 followed by level-1 (denoted as level 1→2), respectively. Although we achieved adequate training accuracy and loss, due to the lack of a good quality dataset, validation accuracy and loss are not satisfactory.

4 Dataset and Result

4.1 Results

The model evaluation takes a lot of time and effort. We were able to produce a good functional model after a lot of trial and error. Behzadan et al. [1] manually labeled the dataset with the four-category system that our classifier had been trained to recognize,

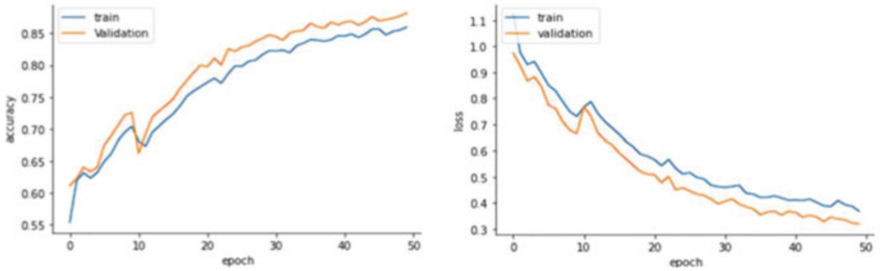


Fig. 5 Training accuracy (left) and loss (right) over the epochs for level-1 classification task

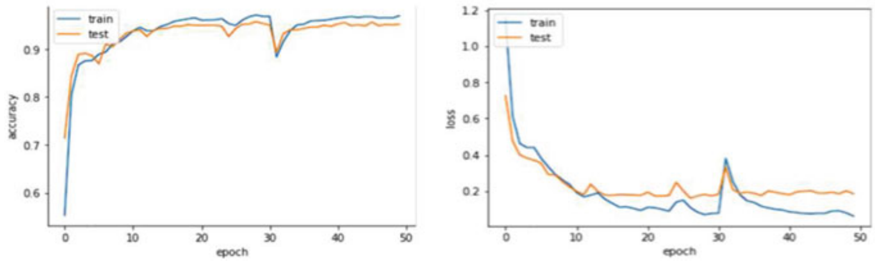


Fig. 6 Training and validation accuracy (left) and loss (right) over the epochs for level-2 classification task

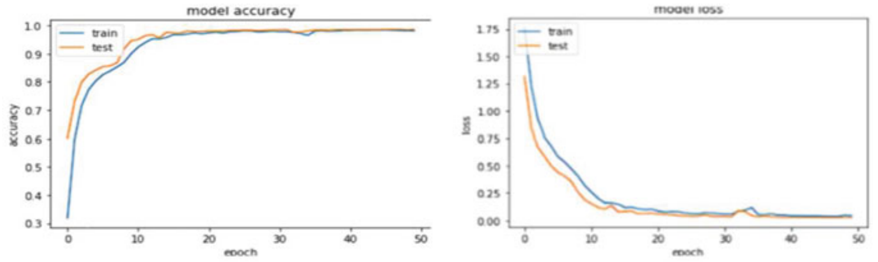


Fig. 7 Training and validation accuracy (left) and loss (right) over the epochs for level 1→2 classification task

'Irrelevant,' 'Business,' 'Threat,' and 'Unknown'. The dataset originally has 21,487 data entries. There are 8351 threats, 4881 irrelevant data, 3967 business/marketing data, and 4288 unknown data among the total. Out of the total threat category tweets, 2094 were DDoS, 372 leak, 1759 general, 1778 vulnerability, 1276 ransomware, 358 botnet, and 714 0day tweets are present. However, a total of 9341 tweets were classified as threats after the level-1 classification was tested on the entire dataset of 21,487 tweets, and they were used as input for the level-2 classifier (see Fig. 2). In both classifiers, we kept the train-test split at 70:30 and the validation split at 20%.

We got 88.08% accuracy on the 30% test data we separated. We got 88.16% test accuracy in level-1 on the whole dataset. This is an acceptable outcome. The level-2 classifier was trained using the 'Threat' predicted tweets from our level-1 classifier. The tweets were classified as follows: 'vulnerability,' 'ransomware,' 'DDoS,' 'leak,' 'General,' '0day,' and 'botnet.' The final accuracy, we achieved in level-2 is **73.26%** (followed by level-1). Detailed result analysis and evaluation are given in the next subsection.

4.2 Result Analysis and Evaluation

Following the effective completion of our job, we were able to obtain a clear picture. Our pipeline threat classifiers were producing positive findings. For the text classification approach, we employed standard evaluation criteria as shown below.

Accuracy: The accuracy with which classifier predictions are made. True Positive and True Negative represent the classifiers' correct predictions.

$$\text{Accuracy} = \frac{(\text{True positive} + \text{True Negetive})}{(\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative})}$$

Precision: The number of positive findings that are correct. It demonstrates how many data instances have been correctly classified and which are true.

$$\text{Precision} = \frac{(\text{True positive})}{(\text{True Positive} + \text{False Positive})}$$

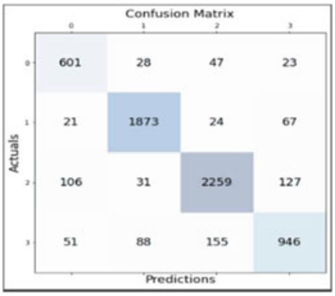
Recall: This is the classifier's correct prediction.

$$\text{Recall} = \frac{(\text{True positive})}{(\text{True Positive} + \text{False Negative})}$$

The test accuracy at level-1 is 88.08% on the test dataset (30% split), and 88.16% on the whole dataset. Figure 8 shows the confusion matrix for the final level-1 classification on the test dataset, which has a precision of 88.13% and a recall of 88.08%.

According to our **proposed method** (see Fig. 2), we were able to achieve 88.14% training accuracy and 86.91% validation accuracy in the level-2 classifier with 9341 tweets classified as 'threat' from level-1, which also contains false positive in another 3 categories, all labeled as 'other' subcategory (means there is no threat involved). The ultimate accuracy, precision, and recall were **81.71%** (the final accuracy of level 1→2), 78.93%, and 81.68%, respectively. The confusion matrix for level 1→2 classifications is shown in Fig. 9. Where 0 denotes 'other,' 1 denotes 'leak,' 2 denotes 'general,' 3 denotes 'vulnerability,' 4 denotes 'DDoS,' 5 denotes 'ransomware,' 6 denotes '0day,' and 7 denotes 'botnet.' 'Other' subcategory is part of

Fig. 8 Level-1 confusion matrix (0 = irrelevant, 1 = business, 3 = threat, 4 = unknown)

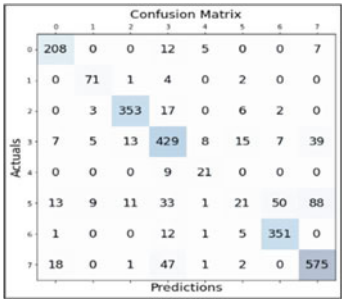


	0	1	2	3
0	601	28	47	23
1	21	1873	24	67
2	106	31	2259	127
3	51	88	155	946

the ‘marketing,’ ‘irrelevant’, and ‘unknown’ categories which are wrongly classified as ‘threat’ category by our level-1 classifier as false positives.

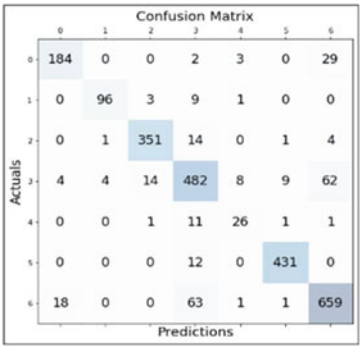
Given the manually separated 8351 threat tweets, we were able to attain 96.90% training accuracy and 95.19% validation accuracy with the level-2 classifier (used standalone). We ended up with a testing accuracy of **90.06%**, precision of 90.41%, and recall of 90.06%. This was a huge advance above previous methods. The level-2 classification’s confusion matrix is shown in Fig. 10. Where 0 denotes a ‘leak,’ 1 denotes a ‘general,’ 2 denotes a ‘vulnerability,’ 3 denotes a ‘DDoS,’ 4 denotes a ‘ransomware,’ 5 denotes a ‘0day,’ and 6 denotes a ‘botnet’.

Fig. 9 Level-1→2 confusion matrix



	0	1	2	3	4	5	6	7
0	208	0	0	12	5	0	0	7
1	0	71	1	4	0	2	0	0
2	0	3	353	17	0	6	2	0
3	7	5	13	429	8	15	7	39
4	0	0	0	9	21	0	0	0
5	13	9	11	33	1	21	50	88
6	1	0	0	12	1	5	351	0
7	18	0	1	47	1	2	0	575

Fig. 10 Level-2 confusion matrix



	0	1	2	3	4	5	6
0	184	0	0	2	3	0	29
1	0	96	3	9	1	0	0
2	0	1	351	14	0	1	4
3	4	4	14	482	8	9	62
4	0	0	1	11	26	1	1
5	0	0	0	12	0	431	0
6	18	0	0	63	1	1	659

Table 2 Performance comparison with a similar system on the same dataset for level-2 classification

Systems	Methodology used	Accuracy (level-1)
Behzadan et al. [1]	CNN	87.56% (binary)
Our proposed Bi-LSTM-based method	BiLSTM	88.08% (multi-class)

4.3 Performance Comparison

Our level-1 classifier finally achieved a threat classification accuracy of 88.08%, while the threat classification done by [1] achieved a test accuracy of 87.56% (see Table 2). At level 1, [1]'s work was more accurate than ours, scoring 87.56% versus 88.08%. However, no proper comparison can be performed because [1] only did binary classification (Threat or not), but we did multi-class classification into four categories in our level-1 classification, and this dataset hasn't been used by any other researchers, so no comparisons can be made. Behzadan et al. [1] did not perform level-2 classification further, therefore no comparison is possible. Table 2 shows the performance comparison.

5 Conclusion

We have successfully made a system model which is better and faster compared to another available system. We used a novel methodology to achieve a better performance. Now to increase the accuracy anymore, more data would be required to train the model. The work will be further extended with the attention mechanism to achieve better accuracy. We are working on adding more features to identify cyber threats in tweets. The code will be publicly available in GitHub. Dionísio et al. [24] in his work has introduced the concept of Multitask Learning which can be implemented in this work in the future. Multitask learning is a concept that makes use of the trained hidden layer of one classification to help increase the accuracy of the next one. This concept is really great. We have found some unique attention parameters which we are already working on to make even more remarkable contributions in this domain hence fulfilling our motto to make the internet a safer place for all. We intend to expand the model's ability to classify threat-containing tweets into additional subcategories in the future, as well as introduce novel attention parameters that will have a stronger impact on this field. Since hackers have stepped up their game, security researchers must also step up their game and put in significant effort to prevent any further crimes from occurring. We're also looking into measures to stop these tweet-based threats from spreading further. As we all know, simply classifying tweets is a task, but it is incomplete unless they are prevented from spreading. As a result, as part of our ongoing research, we're looking into integrating some cybersecurity techniques that can be used to prevent threat tweets from being retweeted or viewed by most people if they violate specific rules.

References

1. Behzadan V, Aguirre C, Bose A, Hsu W (2018) Corpus and deep learning classifier for collection of cyber threat indicators in Twitter stream. In: 2018 IEEE international conference on Big Data (Big Data), pp 5002–5007. https://doi.org/10.1109/BigData.2018.8622506
2. https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.bilstmlayer.html
3. Bose A, Behzadan V, Aguirre C, Hsu WH (2019) A novel approach for detection and ranking of trendy and emerging cyber threat events in Twitter streams. In: Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM '19). Association for Computing Machinery, New York, NY, USA, pp 871–878. https://doi.org/10.1145/3341161.3344379
4. Sceller Q, Karbab E, Debbabi M, Iqbal F (2017) SONAR: automatic detection of cyber security events over the Twitter stream. pp 1–11. https://doi.org/10.1145/3098954.3098992
5. Le B-D, Wang G, Nasim M, Ali Babar M (2019) Gathering cyber threat intelligence from Twitter using novelty classification. pp 316–323. https://doi.org/10.1109/CW.2019.00058
6. Dionísio N, Alves F, Ferreira PM, Bessani A (2019) Cyberthreat detection from Twitter using deep neural networks. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8. https://doi.org/10.1109/IJCNN.2019.8852475
7. Fang Y, Gao J, Liu Z, Huang C (2020) Detecting cyber threat event from Twitter using IDCNN and BiLSTM. Appl Sci 10:5922. https://doi.org/10.3390/app10175922
8. Attarwala A, Dimitrov S, Obeidi A (2017) How efficient is Twitter: predicting 2012 U.S. presidential elections using support vector machine via Twitter and comparing against Iowa Electronic Markets. In: Intelligent systems conference
9. Zong S, Ritter A, Mueller G, Wright E (2019) Analyzing the perceived severity of cybersecurity threats reported on social media. arXiv e-prints
10. Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: Proceedings of the empirical methods in natural language processing
11. Wagner C, Dulaunoy A, Wagener G, Iklody A (2016) MISP: the design and implementation of a collaborative threat intelligence sharing platform. In: Proceedings of the 2016 ACM on workshop on information sharing and collaborative security (WISCS). Association for Computing Machinery
12. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res
13. Sabottke C, Suciu O, Dumitras T (2015) Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In: 24th USENIX Security symposium (USENIX Security 15)
14. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies, vol 1 (long and short papers)
15. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18(5–6):602–610
16. Mikolov T, Chen K, Corrado GS, Dean J (2013) Efficient estimation of word representations in vector space
17. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv e-prints
18. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput

19. Alves F, Ferreira PM, Bessani A (2019) Design of a classification model for a Twitter-based streaming threat monitor. In: 2019 49th annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W)
20. Liu X, He P, Chen W, Gao J (2019) Multi-task deep neural networks for natural language understanding. In: Proceedings of the 57th annual meeting of the Association for Computational Linguistics
21. Baxter J (1997) A Bayesian/information theoretic model of learning to learn via multiple task sampling. Mach Learn 7–39
22. Liao X, Yuan K, Wang X, Li Z, Xing L, Beyah R (2016) Acing the IOC game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security
23. Ruder S, Bingel J, Augenstein I, Søgaard A (2017) Latent multi-task architecture learning. In: Proceedings of the AAAI conference on artificial intelligence
24. Dionísio N, Alves F, Ferreira PM, Bessani A (2020) Towards end-to-end cyberthreat detection from Twitter using multi-task learning. In: 2020 international joint conference on neural networks (IJCNN), pp 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207159

Artificial Neural Network Design for CMOS NAND Gate Using Sigmoid Function

Rupam Sardar, Arkapravo Nandi, Aishi Pramanik, Soumen Bhowmick, De Debashis, Sudip Ghosh, and Hafizur Rahaman

Abstract Artificial Neural Network (ANN) is very useful to predict the future. These predictions can be done in the area of agriculture, transport, finance, health care, etc. Complementary Metal Oxide Semiconductor (CMOS) circuits are used for the design of the hardware. In this paper, we are taking CMOS NAND circuit and experimenting how the NAND gate is useful for an intelligent system. The proposed work was done using the sigmoid function for observing the activation function of NAND gate. The results shown are promising and viable in practical applications.

Keywords Artificial neural network(ANN) · CMOS · Sigmoid function · NAND gate · Aggregation function · Tensorflow

1 Introduction

Modern technologies are using intelligent computations using ANN and Deep Neural Networks (DNN) [3–5] with Artificial Intelligence(AI) [1, 9] which is a vast area for predicting future designing robots [8]. Here we are using the sigmoid activation function for NAND gate truth table realization. Neural Networks [1, 2, 7, 10] are very complex to design and when we are designing the Neural Networks, we must keep in mind that there are input signals and weights that can be multiplied with the input signal and generate the final output after the activation function is applied. In

R. Sardar (✉)
Budge Budge Institute of Technology, Kolkata, West Bengal, India
e-mail: rupamsardar85@gmail.com

A. Nandi
MCKV Institute of Engineering, Liluah, West Bengal, India

D. Debashis
Maulana Abul Kalam Azad University of Technology (MAKAUT), Kolkata, West Bengal, India

A. Pramanik · S. Bhowmick · S. Ghosh · H. Rahaman
Indian Institute of Engineering Science and Technology (IIEST), Shibpur, India
e-mail: soumenbhowmick22@gmail.com

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_9

Truth Table				
A	B	VDD	VSS	Y
0	0	1	0	1
0	1	1	0	1
1	0	1	0	1
1	1	1	0	0

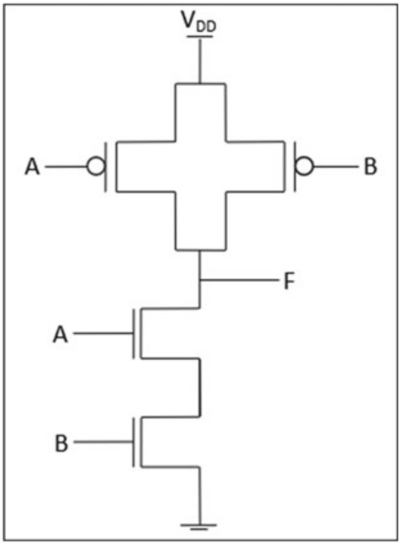


Fig. 1 CMOS NAND gate schematic and truth table

the era of the computer being shaped by AI, NAND gate can be designed in CMOS by two segregations. Pull up can be designed by PMOS and Pull down can be designed by the NMOS. Now NMOS is on when inputs are on or active high and PMOS is on when inputs are active low. We must develop a logic gate that support a neural network. We had applied the voltage of A and B and obtained the output according to the truth table shown in Fig. 1.

Figure 1 shows a CMOS two-input NAND gate. Here, the upper two P-channel transistors are connected in parallel between +Vdd and the output terminal (F) whereas the bottom two N-channel transistors are connected in series between the output terminal (F) and ground. When both of the inputs A and B are logic “0”, then upper two PMOS transistors are “on” and bottom two NMOS transistors are “off”. Hence, output becomes logic “1”.

When both of the inputs A and B are logic “1”, then upper two PMOS transistors are “off” and bottom two NMOS transistors are “on”. Hence, output becomes logic “0”. When one of the inputs is logic “1” and the other is “0”, then the one with logic “0” as the input terminal of the upper PMOS transistors is “on” and the one with logic “0” as the input terminal of the bottom NMOS transistors is “off”. Therefore, the output in both cases is logic “1”.

2 Proposed Methodology

From Fig. 2, the output of the neural network is given by sigmoid activation function:

$$S_{w_1,w_2\ldots w_{n-1},w_n,b}(x_1, x_2 \ldots x_{n-1}, x_n) = \frac{1}{1 + e^{-(w_1x_1+w_2x_2+\cdots+w_{n-1}x_{n-1}+w_nx_n+b)}} \quad (1)$$

Here, in the above equation, $x_1, x_2, x_3, \ldots, x_{n-1}, x_n$ are the n numbers of input features to the neural network [2, 8] present in Fig. 2 and $w_1, w_2, w_3, \ldots, w_{n-1}$, w_n are the n number of input weights associated with input features $x_1, x_2, x_3, \ldots$, x_{n-1}, x_n, respectively, and b is the bias added with summation of weight multiplied with corresponding input features. In Fig. 2, the Pre activation layer computes the aggregation function mentioned in Eq. (1).

In our study, we have applied activation that is sigmoid activation function in activation layer [2] which acts non-linearly. Sigmoid function mentioned in Equation (2) is applied in the models where the output is the prediction of probability belonging to which class. Since the probability of anything exists between 0 and 1, therefore Sigmoid is the right choice.

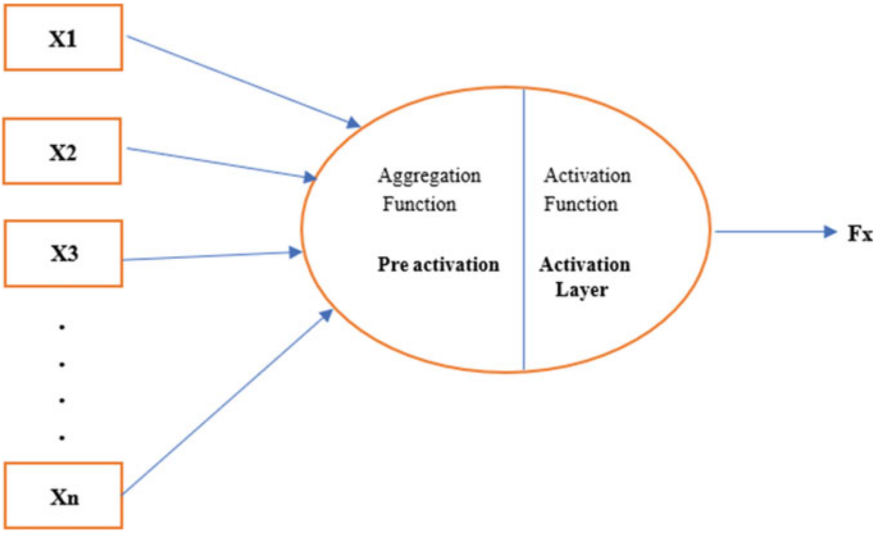


Fig. 2 Diagram of a Neuron in a particular layer

3 Methodology to Design Artificial Neural Network for the CMOS NAND Gate Circuit

After designing the Neural Network in Fig. 3, we obtained several mathematical equations and they are described in this section.

Here, bias is assumed to be 0 in each of the four layers. In the first layer of Fig. 3 of the above model, $Y_1(out)$, $Y_2(out)$, $Y_3(out)$, $Y_4(out)$ are the outputs of each of the four neurons in input layer.

In the input layer,

$$Y_1 = x_1 w_1 + x_2 w_2 + V_{dd} w_3 + V_{ss} w_4 + b_1 \tag{2}$$

$$Y_1(out) = \frac{1}{1 + e^{-Y_1}} \tag{3}$$

$$Y_2 = x_1 w_5 + x_2 w_6 + V_{dd} w_7 + V_{ss} w_8 + b_2 \tag{4}$$

$$Y_2(out) = \frac{1}{1 + e^{-Y_2}} \tag{5}$$

$$Y_3 = x_1 w_9 + x_2 w_{10} + V_{dd} w_{11} + V_{ss} w_{12} + b_3 \tag{6}$$

$$Y_3(out) = \frac{1}{1 + e^{-Y_3}} \tag{7}$$

$$Y_4 = x_1 w_{13} + x_2 w_{14} + V_{dd} w_{15} + V_{ss} w_{16} + b_4 \tag{8}$$

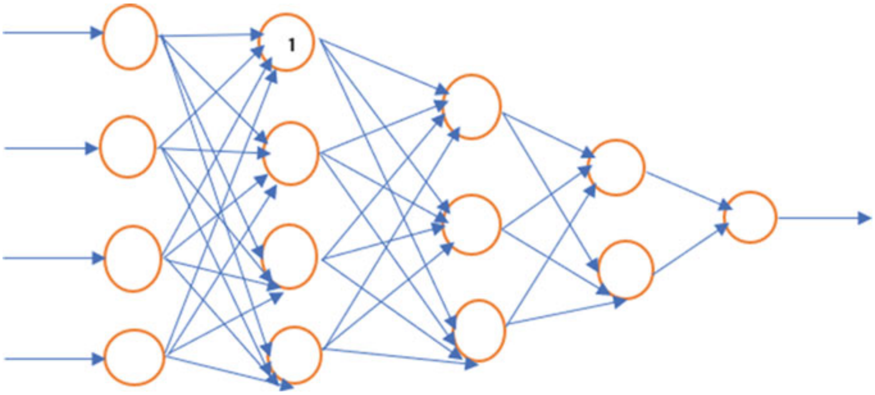


Fig. 3 Representation of the neural network of the CMOS NAND gate

$$Y_4(out) = \frac{1}{1+e^{-Y_4}} \tag{9}$$

In the first hidden (Second) layer,

$$Y_5 = Y_1(out)w_{17} + Y_2(out)w_{18} + Y_3(out)w_{19} + Y_4(out)w_{20} + b_5 \tag{10}$$

$$Y_5(out) = \frac{1}{1+e^{-Y_5}} \tag{11}$$

$$Y_6 = Y_1(out)w_{21} + Y_2(out)w_{22} + Y_3(out)w_{23} + Y_4(out)w_{24} + b_6 \tag{12}$$

$$Y_6(out) = \frac{1}{1+e^{-Y_6}} \tag{13}$$

$$Y_7 = Y_1(out)w_{25} + Y_2(out)w_{26} + Y_3(out)w_{27} + Y_4(out)w_{28} + b_7 \tag{14}$$

$$Y_7(out) = \frac{1}{1+e^{-Y_7}} \tag{15}$$

Similarly, In the second hidden (Third) layer,

$$Y_8 = Y_5(out)w_{29} + Y_6(out)w_{30} + Y_7(out)w_{31} + b_8 \tag{16}$$

$$Y_8(out) = \frac{1}{1+e^{-Y_8}} \tag{17}$$

$$Y_9 = Y_5(out)w_{32} + Y_6(out)w_{33} + Y_7(out)w_{34} + b_9 \tag{18}$$

$$Y_9(out) = \frac{1}{1+e^{-Y_9}} \tag{19}$$

In the last layer (Output Layer),

$$Y_{10} = Y_8(out)w_{35} + Y_9(out)w_{36} + b_{10} \tag{20}$$

$$Y_{10}(out) = \frac{1}{1+e^{-Y_{10}}} \tag{21}$$

In demonstrating the mathematical calculation for simplicity, we have assumed that weight values in each of the four layers are set to 1.

Here, we are taking

$$x_1 = 1, x_2 = 0, V_{dd} = 1 and V_{ss} = 0 \tag{22}$$

as inputs to our proposed neural network in Fig. 3.

$$Y_1 = 1 \times 1 + 1 \times 0 + 1 \times 1 + 1 \times 0 + 0 = 2 \tag{23}$$

$$Y_1(out) = \frac{1}{1 + e^{-2}} = 0.88 \tag{24}$$

$$Y_2 = 1 \times 1 + 1 \times 0 + 1 \times 1 + 1 \times 0 + 0 = 2 \tag{25}$$

$$Y_2(out) = \frac{1}{1 + e^{-1}} = 0.88 \tag{26}$$

since

$$Y_1 = 2 \tag{27}$$

Similarly,

$$Y_3(out) = Y_4(out) = 0.88 \tag{28}$$

$$Y_5 = 1 \times 0.88 + 1 \times 0.88 + 1 \times 0.88 + 1 \times 0.88 + 0 = 3.52 \tag{29}$$

$$Y_5(out) = \frac{1}{1 + e^{-5}} = 0.97 \tag{30}$$

Similarly,

$$Y_6(out) = Y_7(out) = 0.97 \tag{31}$$

In the third layer,

$$Y_8 = 1 \times 0.97 + 1 \times 0.97 + 1 \times 0.97 + 0 = 2.91 \tag{32}$$

$$Y_8(out) = \frac{1}{1 + e^{-8}} = 0.95 \tag{33}$$

Likewise,

$$Y_9(out) = 0.95 \tag{34}$$

On the output layer or last layer,

$$Y_{10} = 1 \times 0.95 + 1 \times 0.95 + 0 = 1.9 \tag{35}$$

$$Y_{10}(out) = \frac{1}{1 + e^{-10}} = 0.87 \tag{36}$$

Therefore, by applying the sigmoid function in the output layer of the network, the final output value becomes equal to 0.87. Here, the output value lies in the range [0,1]. Therefore, we are choosing a certain threshold equal to 0.5. If the output value is greater than this threshold, the proposed neural net will output Y equal to 1 else output would be Y is equal to 0.

Thus, according to the condition mentioned above, Y value is equal to 1. In the mathematical foundation of our proposed methodology,

$$Aggregationfunction : F(x) = x_1w_1 + x_2w_2 + x_3w_3 + \cdots + x_nw_n \tag{37}$$

$$Sigmoidfunction : S(x) = Y_i(out) = \frac{1}{1 + e^{-F(x)}} \tag{38}$$

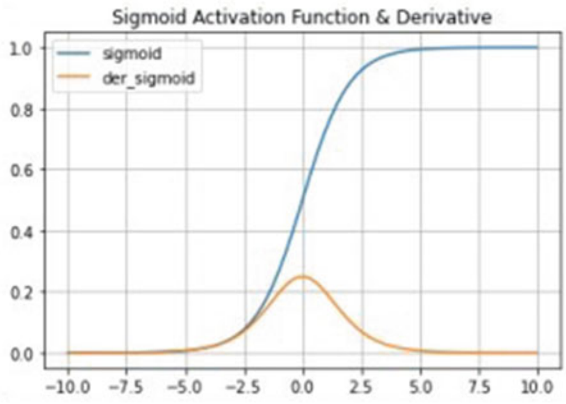
Here Eq. 38 represents Sigmoid Function and Y_i (out) represents the output of the ith neuron [2] in a particular layer. Here i = 1, 2, 3 . . . . . . n

Derivative of the sigmoid function given by

$$S'(x) = S(x)(1 - S(x)) \tag{39}$$

In the Fig. 4, the blue line in the plot represents the plot of sigmoid function curve given by Eq. 38 where the range of values in X-axis lies from −10 to +10 and the Y-axis for the sigmoid function curve ranges from 0 to 1. While the orange plot represents the derivative graph of sigmoid function given by Eq. 39 mentioned above.

Fig. 4 Plot of sigmoid function and its derivative



4 Experimental and Implementation Results with Analysis in Python

In this section, we are describing the details of our proposed neural network model architecture implemented in Python and analyzing the performance of the model on the training and test datasets. In Fig. 3 of the neural network representation of the CMOS Nand gate, it is observed that our proposed model is a 4-layered sequential model architecture comprised of input layer, 2 hidden layers, and output layer.

4.1 Libraries Used

The libraries used in developing the ANN are Numpy, Pandas, Matplotlib, Tensorflow, Scikit-Learn, Seaborn, and Keras. Numpy library has been used in python to perform mathematical and scientific calculations on the input variables of the dataset whereas Pandas is used to load the entire dataset in the python notebook. Tensorflow library is used as “tf” name to create a plot of confusion matrix on the actual output and predicted output by the ANN Model on the test dataset. Keras library served as one of the most important libraries in our model implementation as it is used to create the dense layers of 4-layered ANN architecture and the library has functions that trains the model with optimal parameters, therefore making the model ready to test on the unknown dataset. Scikit-learn which is short called as sk learn is used to split the entire dataset into train and test dataset and the sklearn is used to create metrics to evaluate the performance of our model. Matplotlib library is used to create graphical plots to visualize and analyze the performance of the model on the datasets.

4.2 Dataset Used

The dataset that is used for training and testing performance analysis of our model consists of nine columns:—(i) Input 1 (ii) Input 2 (iii) V_{dd} (iv) V_{ss} (v) Output (vi) Range for Input 1 (vii) Range for Input 2 (viii) Digital Values for Input 1 (ix) Digital Values for Input 2 shown in Table 1. Here Input 1 and Input 2 are the two inputs to the CMOS NAND Gate Circuit which take values as real numbers. While V_{dd} and V_{ss} are the input voltages of the circuit which will take Boolean values as its input i.e., “1” and “0” corresponding to high and low voltages, respectively. The “Output” is the target variable of our ANN model which are of Boolean values in 0 or 1, where 0 volt means “Off” and 1 denotes “On” in digital electronics. The data set that is used for training consists of 20 rows and 5 columns. In both Training and Test Datasets, Input 1 and Input 2 are both analog voltage values. Therefore, we have considered the analog voltage values as 0 V (in digital electronics if the input analog voltage is

Table 1 Instance of training data set of our proposed ANN model

Input 1	Input 2	V_{dd}	V_{ss}	Output	Range for input 1	Range for input 2	Digital values for input 1	Digital values for input 2
0.00	0.00	1	0	1	−11V to +31V	−9V to +20V	0	0
0.00	1.00	1	0	1	−11V to +31V	−9V to +20V	0	1
1.00	0.00	1	0	1	−11V to +31V	−9V to +20V	1	0
1.00	1.00	1	0	0	−11V to +31V	−9V to +20V	1	1
2.00	−3.00	1	1	1	−11V to +31V	−9V to +20V	1	0
−1.00	1.00	0	1	1	−11V to +31V	−9V to +20V	0	1
−8.00	−8.00	0	0	1	−11V to +31V	−9V to +20V	0	0
−10.00	12.00	1	0	1	−11V to +31V	−9V to +20V	0	1
30.56	19.08	1	0	0	−11V to +31V	−9V to +20V	1	1

less than or equal to 0) and as 1 V (in digital electronics if the input analog voltage is greater than 0).

4.3 ANN Model Architecture

The ANN Model is designed using Kerasapi imported from Tensorflow Library in Python. The sequential densely connected neural network model consists of four layers with layer 1 consists of 4 input neurons and 4 output neurons, layer 2 consists of 4 input neurons and 3 output neurons, layer 3 consists of 3 input neurons and 2 output neurons, and layer 4 consisting of 2 input neurons and 1 output neuron. The value of the output neuron in layer 4 (Last Layer) is between 0 and 1. In the mathematical implementation of our proposed model in Fig. 3, sigmoid activation function is used in all the layers 1, 2, 3, and 4 thus while implementing the model in the Python notebook Sigmoid Function is used to analyze the performance of the model. The values from each of the four neurons are fed into the second layer to each of the three neurons, and the value from each of three neurons is fed into two neurons and after then values from two neurons are passed to a single neuron present in the last layer. The last layer is also known as an output layer. Finally, the Sigmoid

Table 2 Architecture of 4-layered Densely connected neural network model

Layer (type)	Output shape	Parameters
dense (Dense)	(None, 4)	20
$dense_1$ (Dense)	(None, 3)	15
$dense_2$ (Dense)	(None, 2)	8
$dense_3$ (Dense)	(None, 1)	3

function present in the output layer gives the value between 0 and 1. The Model of the Sequential Densely Connected [3–5] ANN implemented in Python is shown in Table 2.

Total Parameters: 46.

- Trainable Parameters: 46
- Non-trainable parameters: 0

4.4 Objective of the Model

The problem that we are presenting in this paper is to classify the output of our model [8] as 0 or 1 according to four inputs to the model. So, there are two output classes for the design of the ANN model, i.e., Class 0 (when output of the model is 0 volt) and Class 1 (when the output of the model is 1 volt). Therefore, it is a binary-classification problem.

4.5 Training of the ANN Model

The training dataset consists of 9 rows and 5 columns. The model is compiled using Adam Optimizer which provides the best values of parameters for the neural network and binary crossentropy loss function is used because the problem presented here is binary-classification problem. The metric that is used to measure the performance of the model on both training and testing dataset is Accuracy. The proposed ANN Model is trained for 5000 iterations on the train dataset and gave an accuracy of 100% and loss of 0.0777.

The formula for accuracy is given in Eq. 40:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{40}$$

The plot in Fig. 5 illustrates the performance of the model on the training dataset at different epoch values from 0 to 5000. Epoch means how many times the model is

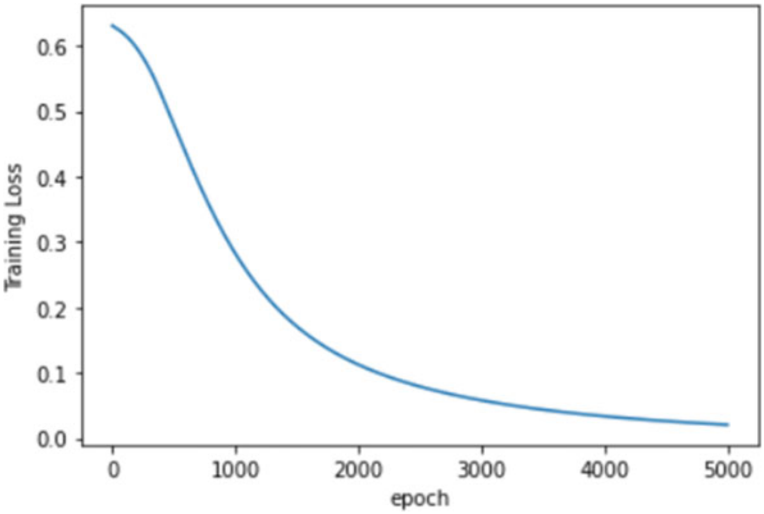


Fig. 5 Plot of training loss of the model versus number of Epochs

trained on the dataset and here at different iterations (epochs) [2, 3, 9], the loss of the model on the training dataset is plotted by using Matplotlib Library in Python. The figure mentioned in Fig. 5 gives us the view that with the increase in the number of training iterations, the loss value on the training dataset gradually decreases.

4.6 Comparison

Here we have shown a comparative table (Table 3) between Our proposed ANN Model of CMOS NAND Gate and CMOS OR Gate design using ANN.

4.7 Performance of the Model on the Test Dataset

Our proposed Neural Network Model on the Test Dataset achieved an accuracy of 80%. The Test Dataset on which the model's performance in evaluated is shown in Table 4.

From Table 4, the output column represents the true or actual outcomes on the test dataset. But, while computing the performance of the model on the test dataset, I have dropped the output column and measured the model's accuracy on the four input features (Input 1, Input 2, V_{dd},V_{ss}). Therefore, the predicted outcomes of the model on the test dataset came out as 0, 0, 0, 1, 0, 1, 1, 1, 0, 0 corresponding to the input row values given in Table 4. From Eq. 40, given, the Accuracy is calculated as 8/10 where 8 is the number of correctly predicted outcomes and 10 is the total number of Outcomes. Thus, model achieves an accuracy of 8/10 i.e., 80% on the test dataset.

4.8 Confusion Matrix

A confusion matrix (CM) is a technique by which the prediction result of a classifier model is summarized on the test dataset. It is the summary of the performance of a classification problem. The matrix represents the ways in which the classification model is confused while making predictions. Moreover, this matrix gives insights about the errors as well as the types of the errors made by the classifier. The structure of CM in Fig. 6 is shown below.

From this Confusion Matrix above in Fig. 6, the following conclusions are made.

1. The model correctly classifies 4 inputs as “Class 0”.
2. 0 inputs in “Class 0” are classified by the model as “Class 1”.

Table 3 Comparison results

Parameters	Our proposed work	[6]
Functions used in each Neuron	Both Aggregate Function and Sigmoid Function used in calculating output values of each Neuron	Only Aggregate Function had been used in calculating output values of each neuron
Weights used in artificial neural network	In our work, we have used arbitrary weight values to give a theoretical implementation of the model. Also, we have implemented the ANN Model using Tensorflow and Keras Library. Here, the tensorflow library takes care in assigning proper weight values on the input edges connecting each neuron during the training phase of the Model	Here, only set of weight values is manually used to give a theoretical implementation of the Model. No Practical Approach had been proposed in this work
Reliability of the model	In our approach, we have created a small dataset manually by using real voltage values and our performance of the model is evaluated on the test dataset. In the test set, our model achieved an accuracy. Therefore, our model is reliable	Whereas here the ANN Model is implemented using mathematical approach only. No practical design has been proposed and so the proposed model has not been tested on real voltage values. So, here the approach becomes less reliable compared to our work

Table 4 Test dataset to evaluate the performance of our proposed ANN model

Input 1	Input 2	V_{dd}	V_{ss}	Output	Range for input 1	Range for input 2	Digital values for input 1	Digital values for input 2
−1.000	1.00	1	0	1	−10V to +100V	−6V to +13V	0	1
99.560	1.00	1	0	0	−10V to +100V	−6V to +13V	1	1
0.000	7.80	1	0	1	−10V to +100V	−6V to +13V	0	1
−9.800	0.00	1	0	1	−10V to +100V	−6V to +13V	0	0
10.430	12.98	1	0	0	−10V to +100V	−6V to +13V	1	1
27.960	−0.98	1	0	1	−10V to +100V	−6V to +13V	1	0
−0.986	0.00	1	0	1	−10V to +100V	−6V to +13V	0	0
11.560	−5.00	1	0	1	−10V to +100V	−6V to +13V	1	0
5.890	1.00	1	0	0	−10V to +100V	−6V to +13V	1	1
9.900	0.56	1	0	0	−10V to +100V	−6V to +13V	1	1

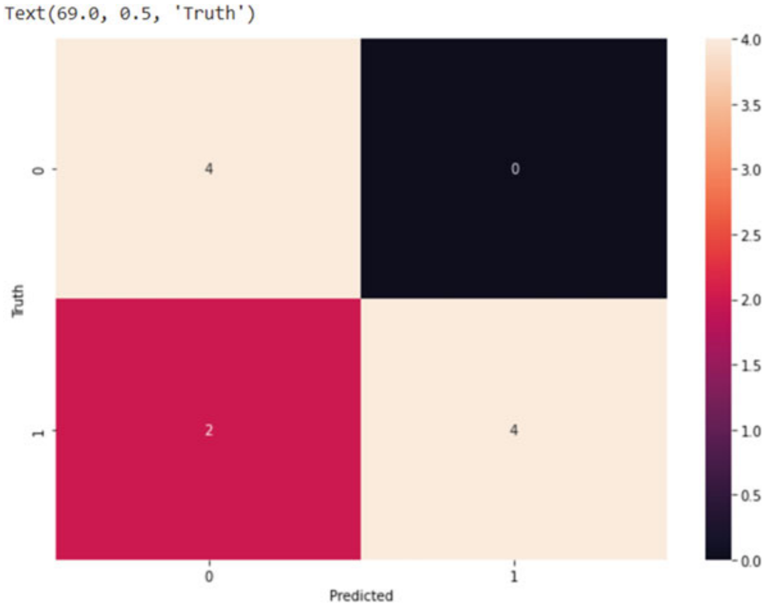
3. 2 inputs in “Class 1”, are misclassified by the model as “Class 0”.
4. The model correctly classifies 4 inputs as “Class 1”.

4.9 *Weights and Biases Present in Each of the 4 Layers of the Model*

By using the model.layers[i].weights command and model.layers[i].bias.numpy() commands we are getting Weight and Bias Matrix [2] in a particular ith layer in Python Notebook. Here model is our proposed neural network model. The values of weights present in the first layer are displayed in the form of (4 × 4) matrix as in the input layer there are four input neurons and four neurons in first layer connected to each of four inputs. The Bias present in the first layer is displayed in the form of 1-dimensional array of (1 × 4) matrix. Thus,

Weight Matrix in First Layer (Input Layer):

[[−0.1936149 , 3.4276466 , 3.504992 , 2.0016828],
[−0.35117793, 1.778528 , 2.8575613, 4.2126145],



```
Text(69.0, 0.5, 'Truth')
```

Fig. 6 Plot of confusion matrix for our proposed artificial neural network model

[2.1481109 ,−1.0041207 ,−1.7299372,−1.3335098],
[−0.31723595, 0.8330988 ,0.6974918 , 0.19749898]]

Bias Matrix in First Layer (Input Layer):

[2.2982867, −1.3996079, −2.033853, −2.1316252]

Similarly, four neurons in the input layer are connected to three neurons in the first hidden layer thereby creating Weight Matrix of size 4 × 3, and since there are three neurons in the layer so bias matrix forms size of 1 × 3.

Weight Matrix in Second Layer (First Hidden Layer):

[[2.830477, −2.9694, −2.2820616],
[−1.3113838, 1.748197, 1.4695135],
[−2.8047054, 2.8619854, 3.2600462],
[−3.3471355, 4.216155, 3.3557246]]

Bias Matrix in Second Layer (First Hidden Layer):

[2.5750675, −3.3858862, −3.8092616]

Likewise,Weight Matrix in Third Layer (Second Hidden Layer):

[[−3.0632641, 3.7905078],
[3.2658846,−3.559395],
[3.6212218, −2.5397708]]

Bias Matrix in Third Layer (Second Hidden Layer):

$$[-2.027568, 1.2913349]$$

Weight Matrix in Fourth Layer (Output Layer):

$$\begin{array}{c}[[-3.9390628],\\ [\ 3.9055228]]\end{array}$$

Bias Matrix in Fourth Layer (Output Layer):

$$[0.333625]$$

5 Conclusion

In this manuscript, we have proposed the design of a small Artificial Neural Network which is designed on a small dataset, and it achieved an 80% accuracy on the test dataset. We will further design a circuit based on the neural network model and will train the model on a large dataset in our future study.

References

1. Chakradhar S, Agrawal V, Bushnell M (1990) Neural net and boolean satisfiability models of logic circuits. IEEE Des Test Comput 7. Accessed 5 Oct 1990
2. Kahraman N, Yildirim T (2008) Technology independent circuit sizing for standard cell-based design using neural networks. www.elsevier.com/locate/dsp. Accessed 13 Dec 2008
3. Khatua K, Maity H, Chattopadhyay S, Sengupta I, Patankar G, Bhattacharya P (2019) A deep neural network augmented approach for fixed polarity AND-XOR network synthesis. In: 2019 IEEE region 10 conference (TENCON 2019)
4. Li Z, Li J, Ren A, Cai R, Ding C, Qian X, Draper J, Yuan B, Tang J (2019) HEIF: highly efficient stochastic computing-based inference framework for deep neural networks. IEEE Trans Comput-Aided Des Integr Circuits Syst 38(8)
5. Lue HT, Hsu PK, Wei ML, Yeh TH, Du PY, Chen WC, Wang KC Optimal design methods to transform 3D NAND flash into a high-density high-bandwidth and low-power nonvolatile computing in memory (nvCIM) accelerator for deep-learning neural networks (DNN)
6. Mandal RK (2016) Design of a CMOS OR gate using artificial neural networks(ANNs). AMSE J 21(1):66–77
7. Masaki A, Hirai Y, Yamada M (1990) Neural networks in CMOS: a case study, July 1990, IEEE Xplore
8. Mohamed AR, Qi L, Wang G (2021) A power-efficient and re-configurable analog artificial neural network classifier. Microelectron J 111:105022
9. Valavala LT, Munot K, Babu RT (2018) Design of CMOS inverter and chain of inverters using neural networks. In: 2018 IEEE international symposium on smart electronic systems
10. Yellamraju S, Kumari S, Girolkar S, Chourasia S, Tete AD (2013) Design of various logic gates in neural networks. In: Annual IEEE India conference, 2013

A Pioneer Image Steganography Method Using the SOD Algorithm

Pabak Indu, Sabyasachi Samanta, and Souvik Bhattacharyya

Abstract Steganography is a technology that has gained popularity in recent years as people's fears and susceptibilities have increased in today's digital environment. As a result of this work, we have shed light on the existing popular techniques of Image Steganography in great detail, and we have also demonstrated a new Spatial Domain Image Steganography technique based on the 'SOD'—Sum-Of-Digit method, wherein secret data is embedded in an image through the use of the proposed algorithm. Moreover, the presented method is compared with existing methodologies on a variety of parameters and found to be efficient and robust, which makes this steganography technique effective. The experimental results can demonstrate the effectiveness and accuracy of the proposed technique in terms of several image similarity metrics, which is a significant benefit.

Keywords Sum of Digits · Spatial method · Steganalysis · Steganography · Ensemble · SRM

1 Introduction

Technological advancements are substantial, and nothing appears to be preventing even our most secret information from falling into the wrong hands. However, our efforts in the field of Cyber-Security have not slowed down when it comes to preventing the breach of personal information. The protection of sensitive information has been a long-standing concern for millennia, and as a result, we have

P. Indu (✉)
Department of Computer Science and Engineering, Adamas University, Kolkata, West Bengal, India
e-mail: pabakindu@yahoo.co.in

S. Samanta
Haldia Institute of Technology, Haldia, West Bengal, India

S. Bhattacharyya
Department of Computer Science and Engineering, University Institute of Technology, University of Burdwan, Burdwan, West Bengal, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_10

continually developed new concepts and strategies to ensure that this information is not compromised as it is transferred from one location to another. Cryptography, Steganography, and watermarking are some of the techniques that are currently in widespread usage. Encryption is a cryptographic technique that takes a message and causes disorder in it or scrambles its arrangement, resulting in a cypher text, and the process is referred to as Encoding. This type of information can only be decoded with the use of a secret key, and the process of obtaining the information is referred to as Decryption. The opponent is aware of the concealed secret information contained within the cypher, and the likelihood of it being deciphered is quite high as a result. In order to protect intellectual property rights in data, watermarking must be used, either with or without the suppression of the presence of communication [1]. Steganography, on the other hand, is intended to conceal the very presence of communication and secret data.

For the past several decades, image steganography has piqued the curiosity of a large number of scientists and academics. If a secret message is intercepted, the primary purpose of image steganography is to conceal the existence of the secret message in order to avoid discovery even if it is discovered. This is accomplished by embedding the secret message inside an innocent cover object (text, audio, video, and picture), which creates the stego object, and then transferring it to the desired destination over a public channel. It is possible that a purposeful or inadvertent obstacle will occur during the transmission process, preventing the message from being correctly transmitted. An optimal steganography strategy should be created in order to preserve an impenetrable stego image, as shown in Fig. 1. The following literatures have a variety of steganography schemes to choose from [2–11]. When it comes to the extraction of these hidden facts, this process is referred to as steganalysis. Almost identical to the steganography idea, with the primary distinction being that it is really a reversed version of the steganography technique.

It is the objective of this research to examine a specific spatial domain picture steganography approach in further detail. Steganalysis is something we come upon

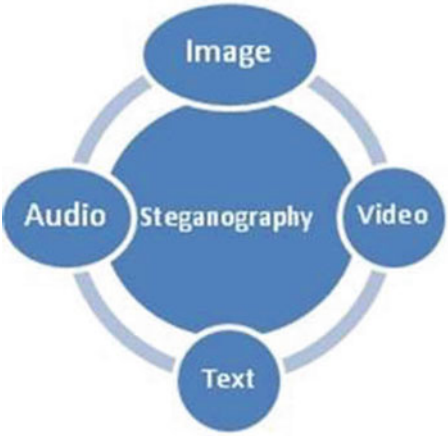


Fig. 1 Types of steganography

later on. Instead of concealing data, steganography is a procedure for detecting hidden data, in which the concealed data is recognised from its cover source, as opposed to steganography. Further, the suggested article is separated into a number of different sections. This paper is divided into five sections: Sect. 2 contains a literature survey on existing models, Sect. 3 describes the proposed model, including an overview of the embedding and extraction algorithms, Sect. 4 presents experimental results on various test images and their corresponding benchmark images, and also compares it with some existing models, and Sect. 5 elaborates the steganalysis results on various image datasets, including comparisons with other models. and concludes with a summary of the proposed paper, which is presented in the form of a conclusion.

2 Review on Existing Methods

Generally, image steganography may be divided into two categories: spatial steganography and transform steganography. The majority of the surveys [12] are concerned with the overall topic of picture steganography. This section discussed well-known picture steganography approaches in the spatial domain that have been utilised in recent years, as well as the emergence of adaptive steganography techniques.

For the most part, this is the most straightforward and well-known option: the least significant bit (LSB) approach, in which data is concealed directly inside the LSB of the pixel values. The development of steganographic technologies necessitated the use of many versions of the existing LSB approach in various bit planes as time went by. Others include adaptive LSB replacement based on several criteria such as edges, texture and brightness of the cover picture estimate the depth of LSB embedding [12–15] and advanced LSB models [12–16], which are described in more detail below.

In addition to the pixel value differencing approach suggested by Wu and Tsai [16], another prominent method is based on pixel value difference (PVD). This is determined by the difference between the values of two adjoining pixels, which determines the number of hidden bits that should be inserted. When the original difference value is not equal to the secret message, the difference values of the two consecutive pixels will be directly adjusted so that their difference values can represent the secret message. However, when the PVD approach adjusts the two successive pixels in order to conceal the secret data in the difference value, a significant degree of distortion may occur in the stego image. A texture picture with a greater resolution can encode more hidden data within the pixel pairs. LSB and PVD have similar payloads, however, PVD has a greater visual imperceptibility and higher visual imperceptibility. The PVD approach avoids the RS detection attacks, but it has a significant disadvantage in that it betrays the existence of a secret message through the use of a histogram. One of the most serious issues is the slipping between the cracks. Furthermore, because general photos have a smooth texture, any secret data will be concealed in the regions with a tiny value, as is the case with most photographs. Many efforts were provided in the literature study that sought to alleviate the constraints of PVD while also enhancing its steganographic aims as a whole. Among them are Multi-Pixel Differencing (MPD) [17], Modulus Function (MF) [18, 19], PVD with LSB [20, 21], block-based PVD [22], and other approaches detailed in [23] and [24]. Other examples include:

Chang et al. [25] propose a unique steganographic approach based on Tri-Way PVD, which is described in detail.

In contrast to the original PVD, the concealing capacity is increased by taking into account three separate directional edges, and the quality distortion is decreased by picking a reference point and applying adaptive algorithms to that point. Dual statistical stego-analysis is used to provide robustness as well as security in this method of study.

Another method, Capacity raising using multi-pixel differencing and pixel-value shifting [17], takes four block pixels and uses the difference between the lowest gray-scale value and the surrounding pixel. The sharpness and smoothness of the embedding are determined using the difference between the lowest gray-scale value and the surrounding pixel. If the difference is significant, it is placed in the sharp block, and if the difference is little, it is placed in the smooth block. When a smooth zone is present, the total embedding capacity diminishes. Following that, pixel-value shifting is performed to improve the overall image quality.

LSB substitution is used in conjunction with a novel approach of modulus function introduced by the Wang et al. Method [26], which is described further below. The main notion of a smooth region is embedded with LSB, and the edges are implanted with the PVD process, with the edges being embedded with LSB. This results in a significant increase in capacity while having no effect on human eyesight.

3 The Proposed Method

This approach is limited to gray-scale photos alone, as the name suggests. The fundamental notion employed in this strategy is the Sum of Digits. We choose a group of bits and modify the pixel in such a way that the sum of its digits equals the decimal value of the pixel we chose. Using this method, we can pick the group so that the change in the pixel value is the smallest possible. The results demonstrate that this innovative method keeps the quality of the image while remaining undetectable by a variety of steganalysis algorithms. This is a decent and acceptable signal-to-noise ratio (PSNR), co-relation, and entropy value for the data we have collected. In the next section, you will find a flow chart that illustrates the embedding and extraction processes in detail, followed by their methodology. Figure 2 shows the embedding process. Figures 3 and 4 show the change in pixel values before and after embedding.

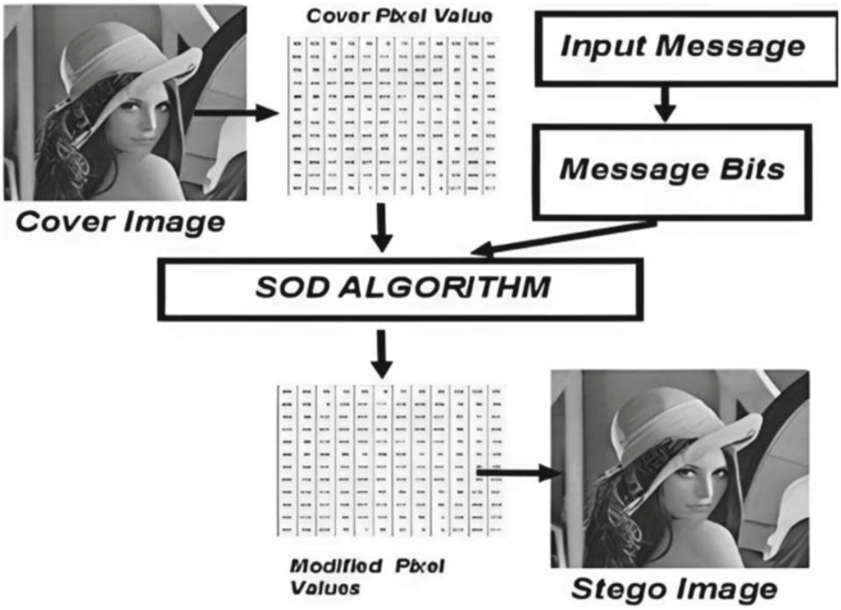


Fig. 2 Block diagram of the embedding algorithm

Fig. 3 Cover image block before embedding

240	123	76
60	135	50
86	55	240

Fig. 4 Cover image block after embedding

240	121	79
60	131	46
90	53	245

3.1 Flow Analysis of the Embedding Algorithm with Following Cases

Step 1: Select the cover image and secret message. Let the Secret Message be ‘AB’.
Step 2: Convert the secret message into binary bits. Thus, the bit pattern we get is

$$\boldsymbol{\beta}_n\boldsymbol{\beta}_{n-1}\boldsymbol{\beta}_{n-2}\boldsymbol{\beta}_{n-3}\boldsymbol{\beta}_{n-4}\boldsymbol{\beta}_{n-5}\boldsymbol{\beta}_{n-}\cdots\boldsymbol{\beta}_7\boldsymbol{\beta}_6\boldsymbol{\beta}_5\boldsymbol{\beta}_4\boldsymbol{\beta}_3\boldsymbol{\beta}_2\boldsymbol{\beta}_1\boldsymbol{\beta}_0$$

Secret Message: AB
‘A’ => 65 => 01000001
‘B’ => 66 => 01000010
So the bit pattern is: 0100000101000010
Step 3: Now take each pixel $\boldsymbol{p}_{ij}$ and find the sum of its digits. Say $\boldsymbol{\rho}_{ij} = \boldsymbol{\varphi}_2 \times 100 + \boldsymbol{\varphi}_1 \times 10 + \boldsymbol{\varphi}_0$, now sum of digits we get is $\sum_{i=0}^{n} \boldsymbol{\varphi}_i = \boldsymbol{\varphi}_{ij}$
76 => Sum of Digits => 13 => 1101
Step 4: Insert $\boldsymbol{\beta}_n$ bits into each pixel in such a way, that $\boldsymbol{\varphi}_{ij}$, is the decimal equivalent of the selected binary bits $\boldsymbol{\beta}_n$. To find the optimal condition in which $\boldsymbol{\beta}_n$ can be embedded into $\boldsymbol{\rho}_{ij}$ we check few cases:
Case 1: If $\boldsymbol{\beta}_n$ consists of 0’s then we convert $\boldsymbol{\rho}_{ij}$, such that $\boldsymbol{\rho}_{newij} = \boldsymbol{\varphi}_2 * 100 + \boldsymbol{\varphi}_1 * 10$ and $\boldsymbol{\rho}_{nextij}$, , $\mu * \eta$ being the resolution of the image, $0 \le \boldsymbol{i} \le \boldsymbol{\mu}$, $0 \le \boldsymbol{j} \le \boldsymbol{\eta}$,

$$\boldsymbol{\rho}_{nextij} = \boldsymbol{\rho}_{i+1j=0};\ \boldsymbol{if}\ \boldsymbol{j} + 1 \ge \boldsymbol{\eta}$$

$$\boldsymbol{\rho}_{nextij} = \boldsymbol{\rho}_{ij+1};\ \boldsymbol{if}\ \boldsymbol{j} + 1 \le \boldsymbol{\eta}$$

$$\boldsymbol{\rho}_{nextij} = \boldsymbol{\varphi}_2 * 100 + \boldsymbol{\varphi}_1 * 10 + \boldsymbol{n}$$

So the new pixels will be: 240 -> 240 and 123 -> 121
$\underline{0}$ 100000101000010
Case 2: Evaluate the value of $\sum_{i=0}^{n} \boldsymbol{\varphi}_i = \boldsymbol{\varphi}_{ij}$, insertion of $\boldsymbol{\beta}_n$ into $\boldsymbol{\rho}_{ij}$ depends on$\boldsymbol{\varphi}_{ij}$, we take nearest value of $\boldsymbol{\varphi}_{ij}$ as $\boldsymbol{\beta}_n$. We calculate the $\boldsymbol{min}_{\partial}$, for this we start from the $\boldsymbol{\beta}_n^{th}$ bit and move forward.

$$\partial_0 = \boldsymbol{\varphi}_{ij} - \left(\boldsymbol{\beta}_n\right)_{10}\partial_1$$

$$\partial_1 = \boldsymbol{\varphi}_{ij} - \left(\boldsymbol{\beta}_n\boldsymbol{\beta}_{n-1}\right)_{10}$$

$$\partial_2 = \boldsymbol{\varphi}_{ij} - \left(\boldsymbol{\beta}_n\boldsymbol{\beta}_{n-1}\boldsymbol{\beta}_{n-2}\right)_{10}$$

$$\partial_3 = \boldsymbol{\varphi}_{ij} - \left(\boldsymbol{\beta}_n\boldsymbol{\beta}_{n-1}\boldsymbol{\beta}_{n-2}\boldsymbol{\beta}_{n-3}\right)_{10}$$

$$\partial_4 = \varphi_{ij} - \left(\boldsymbol{\beta}_n \boldsymbol{\beta}_{n-1} \boldsymbol{\beta}_{n-2\boldsymbol{\beta}_{n-4}\boldsymbol{\beta}_{n-5}}\right)_{10}$$

$$\boldsymbol{min}_{\partial} = \mathbf{min}(\partial_4, \partial_3, \partial_2, \partial_1, \partial_0)$$

$\partial_{\mathbf{n}} = \boldsymbol{min}_{\partial}$
$\varphi_{newij} = \boldsymbol{min}_{\partial}$,
So we have to choose $\boldsymbol{\rho}_{newij}$ such that
$\boldsymbol{\rho}_{newij} = \varphi_{new2} \times 100 + \varphi_{new1} \times 10 + \varphi_{new0}$,
And,
$\varphi_{newij} = \sum_{i=0}^{3} \varphi_{newi}$.
Nearest is 10000 which is the next set of 5 bits.
76 => 1101 => 10000 => (16) => 79
0 10000 0101000010
Step 5: The Remaining bit stream is

$$\boldsymbol{\beta}_{n-1}\boldsymbol{\beta}_{n-2}\boldsymbol{\beta}_{n-3}\boldsymbol{\beta}_{n-4}\boldsymbol{\beta}_{n-5}\boldsymbol{\beta}_{n-} \cdots \boldsymbol{\beta}_7\boldsymbol{\beta}_6\boldsymbol{\beta}_5\boldsymbol{\beta}_4\boldsymbol{\beta}_3\boldsymbol{\beta}_2\boldsymbol{\beta}_1\boldsymbol{\beta}_0$$

Step 6: Now repeat 3–5 for remaining pixels.

3.2 Flow Analysis of the Extraction Algorithm with Following Cases

Step 1: Select the stego image.
Step 2: For extracting we follow the criteria, such as:
Case 1: If $\mathbf{mod}(\boldsymbol{\rho}_{ij}, 10) = 0$, $\boldsymbol{\beta}_n$ Consists of 0's, and n $= \boldsymbol{mod}(\boldsymbol{\rho}_{nextij}, 10)$, $\boldsymbol{\beta}_n = 000\ldots\boldsymbol{nterms}$
240 => 0's
121 => 1 => 0
0
Case 2: If $\mathbf{mod}(\boldsymbol{\rho}_{ij}, 10) \neq 0$, represent it as
$\sum_{i=0}^{n} \varphi_i = \varphi_{ij}$, where
$\boldsymbol{\rho}_{ij} = \varphi_2 \times 100 + \varphi_1 \times 10 + \varphi_0 \partial_n = \varphi_{ij} \boldsymbol{\beta}_n = \boldsymbol{bin}(\partial_n)$
79 => 7 + 9 => 16 => 10000
0 10000
Step 3: Evaluate all such $\boldsymbol{\beta}_n$ values for all the pixel values in the image.
Step 4: All the $\boldsymbol{\beta}_n$ evaluated from the pixel values are represented as a whole, such as, $\boldsymbol{\beta}_n\boldsymbol{\beta}_{n-1}\boldsymbol{\beta}_{n-2}\boldsymbol{\beta}_{n-3}\boldsymbol{\beta}_{n-} \cdots \boldsymbol{\beta}_4\boldsymbol{\beta}_3\boldsymbol{\beta}_2\boldsymbol{\beta}_1\boldsymbol{\beta}_0$.
01000001 01000010 11
Step 5: Divide them into group of $\boldsymbol{\beta}_8$, and then convert them into their equivalent character form.
01000001 => 65 => A
01000010 => 66 => B

Step 6: Repeat step 2–5 for all message bits.

4 Experimental Results and Analysis

There is a thorough explanation of the experimental analysis that was performed using this approach included in this document. According to certain current approaches, we assess and compare both the original and stego photos in this paper. In order to test the findings, the ‘lena.pgm’ cover picture, as shown in Fig. 5, is utilised. The sizes of the photos are determined at random. It is decided on the usage of a sequence of randomly generated digits or characters as the secret message to be placed into the cover graphics. The peak signal-to-noise ratio (PSNR) was used to assess the overall picture quality of the image. The probability of a signal being received is defined as follows:

$$\mathrm{PSNR} = 10.\log_{10}\frac{\left(2^{\mathrm{B}}-1\right)^{2}}{\mathrm{MSE}}\mathrm{PSNR} = 10.\log_{10}\frac{\left(2^{\mathrm{B}}-1\right)^{2}}{\mathrm{MSE}}\mathrm{dB}$$

And

$$\mathrm{MSE} = \frac{1}{\mu \times \eta}\sum_{i=0}^{\mu-1}\sum_{j=0}^{\eta-1}\left(\gamma_{ij} - \ddot{\Upsilon}_{ij}\right)^{2}.$$

As shown in this illustration, is the cover image pixel with the coordinates (*ij*), and is the stego-image pixel with the same coordinates. The greater the PSNR number, the greater the likelihood that the difference between the cover picture and the stego image is undetectable by human eyes. Table 1 displays the results of the experiments conducted on a variety of typical cover pictures.

As we can see in Table 1, there is little question that the stego-image quality created by the suggested technique was pretty good, with no discernible difference

Fig. 5 Visual effects of proposed steganography scheme **a** PSNR 58.8 dB, **b** PSNR 52.8 dB, **c** PSNR 48.4 dB, **d** PSNR 37.58 dB

Table 1 Experimental result

Image	Length of embedding (bpp)	PSNR (dB)	Correlation	Entropy
lena512.bmp (512 × 512)	327	60.492	0.999987	7.445684
	3270	50.105	0.999863	7.445894
	8192	46.075	0.999660	7.441868
	16,384	43.085	0.999342	7.426170
	32,768	40.065	0.999034	5.981834
zelda512.bmp (512 × 512)	327	60.260	0.999981	7.267064
	3270	50.094	0.999808	7.266713
	8192	46.042	0.999523	7.262412
	16,384	43.066	0.999086	7.245796
	32,768	40.025	0.998766	7.359929
Cameraman.bmp (512 × 512)	327	53.754	0.999971	6.910685
	3270	44.014	0.999738	6.920358
	8192	40.019	0.999391	6.874710
	16,384	37.098	0.998964	6.565247
	32,768	36.617	0.998872	6.443998
barbara.pgm (512 × 512)	327	59.809	0.999988	7.632077
	3270	50.053	0.999893	7.631960
	8192	46.027	0.999736	7.627175
	16,384	42.944	0.999480	7.607857
	32,768	39.975	0.999035	7.532511
boat.512.tiff (512 × 512)	327	59.573	0.999983	7.191792
	3270	50.125	0.999856	7.193764
	8192	46.135	0.999647	7.193725
	16,384	43.074	0.999307	7.184372
	32,768	40.090	0.998712	7.143109

between the original cover picture and the final product. Furthermore, as shown in Table 2, embedding has a greater capacity than the other approaches now in use. In addition, we analyse the payload capacity of the suggested scheme in order to determine the overall quality of the technique. When it comes to payload capacity, it is defined as the ratio of the number of embedded bits to the number of cover bits in a message. It is denoted by the symbol.

$$\mathtt{C} = \frac{\text{number of } \beta_n}{\mu \times \eta}$$

$$\mathtt{C}_{\text{avg}} = 2.54\,\text{bits/pixel}$$

Table 2 Comparison of embedding capacity with existing method

Existing solutions	Cover image	Capacity (bytes)	PSNR (dB)
Wu and Tsai original PVD [16]	Lena	50,960	41.79
	Baboon	56,291	37.90
	Pepper	50,685	40.97
	Jet	51,243	40.97
Wang et al.'s method [26]	Lena	51,226	46.96
	Baboon	57,138	43.11
	Pepper	50,955	46.10
	Jet	51,234	46.19
Yang and Weng's method [17]	Lena	73,814	35.98
	Baboon	78,929	33.17
	Pepper	74,280	34.79
	Jet	73,001	33.89
Chang et al.'s method [27]	Lena	76,170	40.80
	Baboon	82,672	32.63
	Pepper	75,930	40.25
	Jet	76,287	38.46
Chang et al.'s tri-way PVD [25]	Lena	75,836	38.89
	Baboon	82,407	38.93
	Pepper	75,579	38.50
	Jet	76,352	38.70
SOD method (proposed method)	Lena	78,000	36.43
	Baboon	75,000	36.16
	Pepper	75,000	36.03
	Jet	65,000	36.05

Table 2 compares and contrasts the suggested technique with an alternative algorithm. Because the PSNR value is higher than that of the previous algorithm, we may conclude that it is superior to the existing Yang and Weng's technique. The concessions made in terms of image quality are negotiated in light of the unpredictable nature of the medium.

In current times, steganalysis is performed in two stages: first, the image models are extracted, and then the image models are trained using a machine learning tool to discriminate between the cover picture and the stego image. The training is carried out with the use of appropriate picture models, which allow for the differentiation of each and every cover and stego image.

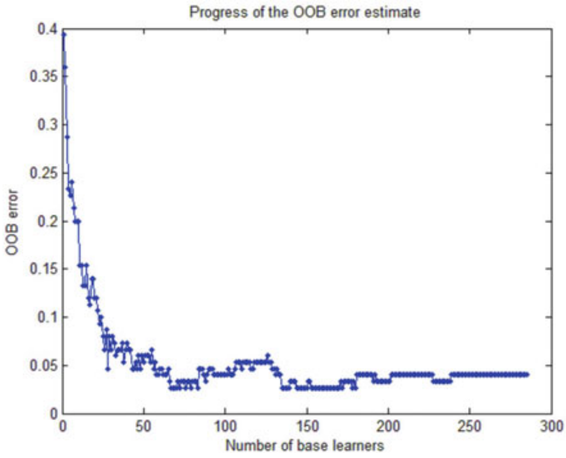
We employed Rich models for steganalysis of digital pictures [28] for the extraction phase, and the procedure begins with the assembly of distinct noise components retrieved from the submodels produced by neighbourhood sample models. The primary goal is to discover various functional connections between nearby pixel blocks in order to recognise a diverse range of embedding models. In the approach, we employed a straightforward way to produce the submodels, which we then combined into a single feature before developing a more complicated submodel and classifying them. Both the cover picture and stego image are used in the extraction procedure, and the attributes of each are extracted.

During the training phase, we begin by dividing the picture database into sets, with each set including the same number of matching stego images as well as associated cover images. In order to do this, we employed the ensemble classifier, as described in [29] and [30]. The ensemble classifier is made up of L binary classifiers, which are referred to as the basis learners.

1, 2, 3,…, L, each trained on a distinct submodel of a randomly selected feature space. In order to evaluate the performance detection of the submodel on an unknown data set, it is convenient to use the out-of-band error estimate. The rich model is then assembled using the OOB error estimates from each submodel in the following step. Figure 6 depicts the progress made in estimating out-of-band error estimates in the proposed model. Following the generation of the OOB estimations [31], the implementation of a Receiver Operating Characteristic (ROC) graph is carried out, which conveys the categorisation of the models into stegogramme and non-stegogramme classes. The ROC curve of 50 stegogrammes and non-stegogrammes from our picture dataset on different embeddings of 0.01, 0.1, and 0.25 bpp is depicted in Fig. 7. The embeddings with a bpp of less than 0.25 are acceptable and are not identified by the model (Fig. 8).

It is noticed that the suggested technique outperforms the well-known Tri-Way PVD when the ROC curves for Tri-Way PVD and the proposed method are shown at an embedding rate of 0.25 bps. When comparing the two methods at an embedding rate of 0.25 bpp, the ensemble recognises the Tri-Way PVD completely, but the suggested approach is only partially identified by the ensemble.

Fig. 6 OOB error estimates



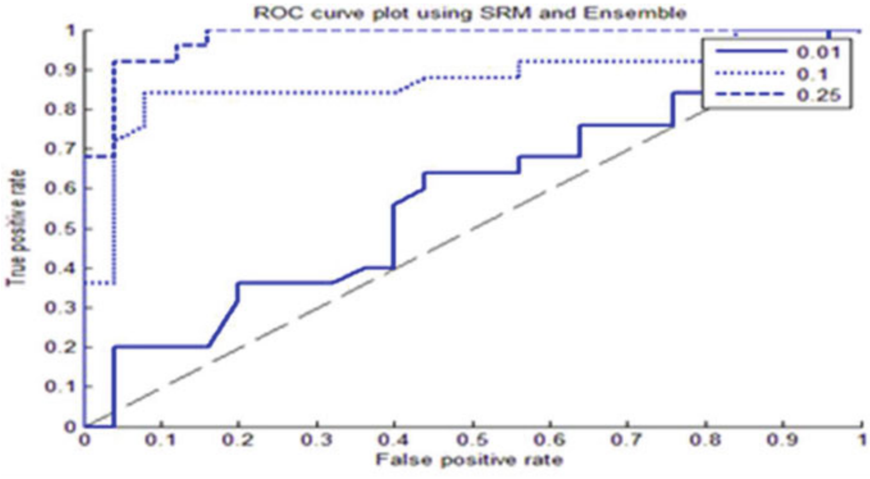
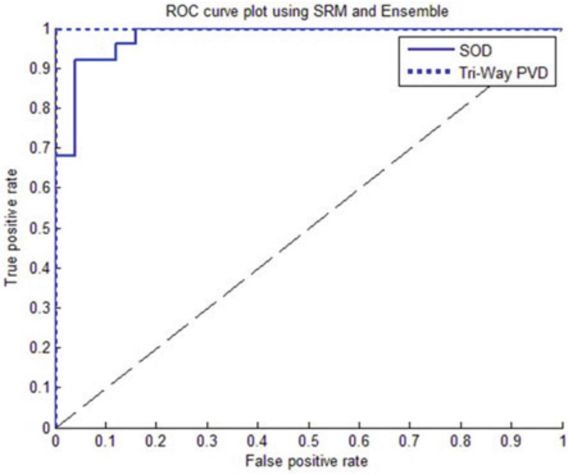


Fig. 7 ROC curve embedding at 0.01, 0.1, and 0.25 bpp

Fig. 8 ROC curve comparison at 0.25 bpp



5 Conclusions

With the help of algorithms, figures, and examples, this proposed method has shed light on the various techniques available for implementing Steganography and has narrowed its focus to our own devised method of Image Steganography, which explains how to embed a message into any cover image in a clear and understandable manner. Because it has been tested and run several times, the novel technique is one-of-a-kind and has shown to be quite reliable thus far. The results of the steganalysis are pretty satisfactory and comparable to those of other models. Moreover, the application of SRM steganalysis is investigated, and it is discovered that the embedding is pretty acceptable below 0.25 bpp when displaying its ROC curve with an ensemble classifier.

References

1. Holub V, Fridrich J, Denemark T (2014) Universal distortion function for steganography in an arbitrary domain. EURASIP J Inf Secur 2014(1):1
2. Balasubramanian C, Selvakumar S, Geetha S (2014) High payload image steganography with reduced distortion using octonary pixel pairing scheme. Multimed Tools Appl 73(3):2223–2245
3. Zhang X, Wang S (2006) Efficient steganographic embedding by exploiting modification direction. IEEE Commun Lett 10(11):781–783
4. Liao X, Qin Z, Ding L (2017) Data embedding in digital images using critical functions. Signal Process: Image Commun 58:146–156
5. Sun H-M et al (2011) Anti-forensics with steganographic data embedding in digital images. IEEE J Sel Areas Commun 29(7):1392–1403
6. Kieu TD, Chang C-C (2011) A steganographic scheme by fully exploiting modification directions. Expert Syst Appl 38(8):10648–10657
7. Kuo W-C, Kuo S-H, Huang Y-C (2013) Data hiding schemes based on the formal improved exploiting modification direction method. Appl Math Inf Sci Lett 1(3):1–8
8. Tsai P, Hu Y-C, Yeh H-L (2009) Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process 89(6):1129–1143
9. Chen N-K et al (2016) Reversible watermarking for medical images using histogram shifting with location map reduction. In: 2016 IEEE international conference on industrial technology (ICIT). IEEE
10. Pan Z et al (2015) Reversible data hiding based on local histogram shifting with multilayer embedding. J Vis Commun Image Represent 31:64–74
11. Al-Dmour H, Al-Ani A (2016) A steganography embedding method based on edge identification and XOR coding. Expert Syst Appl 46:293–306
12. Qazanfari K, Safabakhsh R (2014) A new steganography method which preserves histogram: generalization of LSB++. Inf Sci 277:90–101
13. Tavares JRC, Junior FMB (2016) Word-hunt: a LSB steganography method with low expected number of modifications per pixel. IEEE Lat Am Trans 14(2):1058–1064
14. Tseng H-W, Leng H-S (2014) High-payload block-based data hiding scheme using hybrid edge detector with minimal distortion. IET Image Proc 8(11):647–654
15. Nguyen TD, Arch-Int S, Arch-Int N (2016) An adaptive multi bit-plane image steganography using block data-hiding. Multimed Tools Appl 75(14):8319–8345
16. Wu D-C, Tsai W-H (2003) A steganographic method for images by pixel-value differencing. Pattern Recogn Lett 24(9–10):1613–1626
17. Yang C-H, Wang S-J, Weng C-Y (2010) Capacity-raising steganography using multi-pixel differencing and pixel-value shifting operations. Fundam Inform 98
18. Pan F, Li J, Yang X (2011) Image steganography method based on PVD and modulus function. In: 2011 international conference on electronics, communications and control (ICECC). IEEE
19. Liao X, Wen Q, Zhang J (2013) Improving the adaptive steganographic methods based on modulus function. IEICE Trans Fundam Electron Commun Comput Sci 96(12):2731–2734
20. Wu H-C et al (2005) Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEE Proc-Vis, Image Signal Process 152(5):611–615
21. Jung K-H (2010) High-capacity steganographic method based on pixel-value differencing and LSB replacement methods. Imaging Sci J 58(4):213–221

22. Yang C-H et al (2011) A data hiding scheme using the varieties of pixel-value differencing in multimedia images. J Syst Softw 84(4):669–678
23. Hussain M et al (2015) Pixel value differencing steganography techniques: analysis and open challenge. In: 2015 IEEE international conference on consumer electronics-Taiwan. IEEE
24. Hussain M et al (2017) A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement. Signal Process: Image Commun 50:44–57
25. Chang K-C et al (2008) A novel image steganographic method using tri-way pixel-value differencing. J Multimed 3(2)
26. Wang C-M et al (2008) A high quality steganographic method with pixel-value differencing and modulus function. J Syst Softw 81(1):150–158
27. Chang K-C et al (2007) Image steganographic scheme using tri-way pixel-value differencing and adaptive rules. In: Third international conference on intelligent information hiding and multimedia signal processing (IIH-MSP 2007), vol 2. IEEE
28. Kodovsky J, Fridrich J, Holub V (2011) Ensemble classifiers for steganalysis of digital media. IEEE Trans Inf Forensics Secur 7(2):432–444
29. Fridrich J et al (2011) Steganalysis of content-adaptive steganography in spatial domain. In: International workshop on information hiding. Springer, Berlin, Heidelberg
30. Kodovský J, Fridrich J (2011) Steganalysis in high dimensions: fusing classifiers built on random subspaces. In: Media watermarking, security, and forensics III, vol 7880. International Society for Optics and Photonics
31. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur 7(3):868–882

Leveraging Potential of Deep Learning for Remote Sensing Data: A Review

Kavita Devanand Bathe and Nita Sanjay Patil

Abstract Remote sensing has witnessed impressive progress of computer vision and state of art deep learning methods on satellite imagery analysis. Image classification, semantic segmentation and object detection are the major computer vision tasks for remote sensing satellite image analysis. Most of work in literature is concentrated on utilization of optical satellite data for the aforementioned tasks. There remains a lot of potential in usage of Synthetic Aperture Radar (SAR) data and its fusion with optical data which is still at its nascent stage. This paper reviews, state of the art deep learning methods, recent research progress in Deep learning applied to remote sensing satellite image analysis, related comparative analysis, benchmark datasets and evaluation criteria. This paper provides in depth review of satellite image analysis with the cutting edge technologies and promising research directions to the budding researchers in the field of remote sensing and deep learning.

Keywords Computer vision · Deep learning · Synthetic aperture radar · Semantic segmentation

1 Introduction

Remote sensing (RS) plays vital role in earth observation. The RS technology utilizes airborne sensors, space borne sensors and other platforms for data acquisition. Generally, the space borne satellite sensors due to its large special coverage and frequency visits, offer an efficient way to observe the earth surface and its changes on daily

K. Devanand Bathe (✉) · N. S. Patil
Datta Meghe College of Engineering, Airoli, Navi Mumbai, India
e-mail: kavitag@somaiya.edu

N. S. Patil
e-mail: nita.patil@dmce.ac.in

K. Devanand Bathe
K J Somaiya Institute of Technology, Mumbai, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_11

basis. Synthetic Aperture Radar (SAR) and optical imagery are widely used modalities for these space borne sensors. Rapid development of RS technology improves the temporal, spatial and spectral resolution of satellite imagery acquired from these modalities. The advancement in earth observation technologies provide enormous amount of satellite data. Optical satellite imagery is acquired from satellites like QuickBird, Landsat, Satellite Pour l'Observation de la Terre (SPOT), Moderate Resolution Imaging Spectroradiometer (MODIS) and the recently launched Sentinel 2. It provides optical data which is successfully utilized for variety of remote sensing applications. The optical data is easy to interpret; however, it suffers from cloud cover problem. The optical sensors are incapable of acquiring data in bad weather conditions and during night time. Synthetic aperture radar (SAR) is widely adapted to overcome the limitations of optical data as SAR can not only penetrate through clouds but also provide data in all weather conditions irrespective of day and night. Space borne radar sensors like SEASAT, SIR-A, SIR-B, ERAS-1, JERS-1, ERAS, RADARSAT, SRTM ALOS POLSAR, COSMO-SKYMed, RISAT and the recently launched Sentinel-1 provide information which can be utilized for earth observation. The satellite imagery is complex in nature. It is analyzed for various tasks such as image scene classification, object detection and semantic segmentation. Information extraction, analysis, machine interpretation from these satellite images is found to be difficult in contrast to RGB images. In the past few years, researchers have explored Rule based methods, Statistical methods and Machine learning based methods for interpretation of satellite imagery. The conventional rule based remote sensing methods utilize spectral indices for optical imagery and radar backscatter intensity values (σ0) for SAR imagery. Apart from conventional remote sensing methods, machine learning based methods like support vector machine (SVM) [1], random forest (RF) [2], decision tree (DT), K-nearest neighbor (KNN), KMEANS clustering and iterative self-organizing data analysis technique ISODATA are commonly adapted in literature to solve various computer vision tasks pertaining to optical and SAR remote sensing. These methods have shown promising results on remote sensing data for several applications like Flood mapping and Land use Land cover. The aforementioned techniques rely heavily on manual image feature extraction. They are time consuming and need human intervention [3, 4]. The accuracy and performance of these methods depend on expertise of remote sensing analyst [5]. This leads to delay in the decision making process specifically in case of disasters like flood or landslide where an immediate release of maps is of prime importance for saving lives and property.

In recent years, Deep learning (DL)—a sub domain of machine learning has emerged as an effective tool. In contrast to traditional methods, DL can automatically extract image features [6, 7]. It is successfully used in a wide variety of applications including remote sensing. The promising results of deep learning methods on remote sensing tasks seek widespread attention of researchers in the remote sensing community. Several research articles have been published pertaining to the usage of deep learning for remote sensing tasks and related applications. The following study comprises of relative records of the past few years. The analysis shows that few articles are published on generalized topics of which few are task oriented. There exists

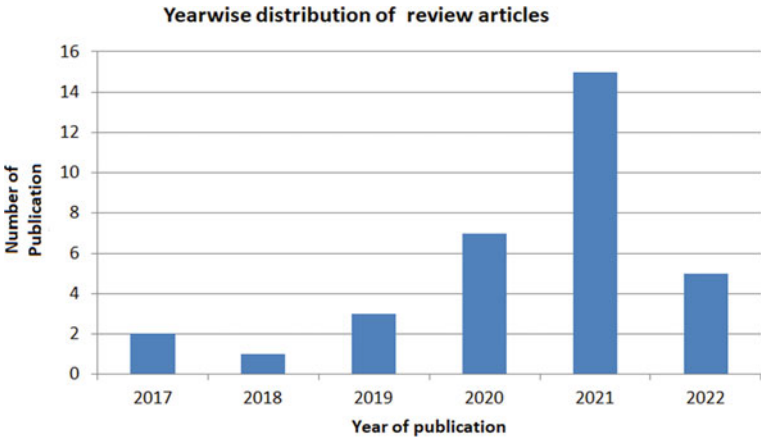


Fig. 1 Year wise distribution of published review articles

another category that is specific to optical data while others are focusing on SAR data. Though many research articles are available, comprehensive review of Optical, SAR remote sensing and its fusion with Deep learning is under explored and is the motivation behind this paper.

The contributions of this paper are threefold which are as follows:

1. This paper provides comprehensive review and comparative analysis of major remote sensing tasks where deep learning can be implemented on satellite imagery.
2. This paper summarizes various benchmark datasets available for optical data, SAR data and optical-SAR fusion based data.
3. It discusses the related challenges and potential future directions for usage of deep learning on satellite imagery.

Figure 1 depicts the distribution of published literature in the past few years.

The organization of this paper is as follows:

Section 2 gives an overview of recent development in Deep learning methods. Section 3 describes the research progress of deep learning for remote sensing whereas in Sect. 4, remote sensing benchmark datasets are illustrated. In Sect. 5 evaluation criteria is depicted. Finally, in Sect. 6 conclusion and future direction for researchers are revealed.

2 Recent Development in Deep Learning

In recent years, Deep learning has significantly contributed to a variety of computer vision tasks like classification, segmentation, object detection and scene understanding. The salient feature of deep learning methods is that they are capable of learning the image features automatically. This feature of deep learning methods makes it highly compatible to remote sensing satellite imagery. Deep learning is categorized as Discriminative learning, Generative learning and Hybrid learning. These Deep neural network categories are discussed in detail in the subsequent sections.

2.1 Deep Networks for Discriminative Learning

The learning approach in which training data comprises of input and its corresponding target label falls under the category of discriminative learning. Convolution neural networks (CNNs), Recurrent Neural Networks (RNN) and its variants such as Long short term memory (LSTM), Gated recurrent unit (GRU), Bidirectional LSTM are normally adapted for remote sensing tasks. The following section describes the aforementioned deep learning algorithm in detail.

Convolutional neural networks (CNN)

Convolutional neural network is a popular supervised deep learning architecture that is widely employed to solve the problems pertaining to visual imagery and audio signals [8]. The strength of this technique lies in the fact that it extracts low level features and high level features from raw input data in contrast to traditional machine learning techniques for manual feature extraction. Convolution layer, pooling layer and fully connected layer are the major building blocks of this architecture. Convolution layer plays a vital role in feature extraction. The base of the convolution layer is the convolution operation that leverages sparse interactions, parameter sharing and equivariant representations [9]. Typically, convolution layer includes hyper parameters as number of kernels, size of the kernel, stride and padding. Convolution layer is followed by pooling layer that subsamples the feature map generated in the previous layer using pooling operation. Min pooling, max pooling, average pooling and global average pooling are the commonly adapted pooling operations. In the Fully Connected Layer, the output feature map of pooling layer is flattened and mapped to an one dimensional array. This in turn is connected to the dense layer in which all neurons from one layer are connected to the neurons in other layer. Each node in the output layer represents the probability of node with reference to the classes [10].

Activation function is used in CNN in different layers. Popularly used activation function in various layers in CNN and other networks include Sigmoid, ReLU, Leaky Relu, and softmax. Mathematical equations of aforementioned activation functions are represented from equation number 1 to 4 respectively.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

$$f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \tag{2}$$

$$f(x) = \begin{cases} x, x > 0 \\ 0.001x, x \leq 0 \end{cases} \tag{3}$$

$$f(x) = \frac{exp(xi)}{\sum_{i=1}^{n} exp(xi)} \tag{4}$$

Over the years several modern CNN architectures [11] are utilized for various remote sensing tasks. Remote sensing community leveraged the potential of CNN on optical and SAR imagery using different approaches. The promising results of CNN on various remote sensing tasks like classification, object detection and segmentation create new pathway to explore the new areas of remote sensing [12].

Recurrent neural networks (RNN)

Recurrent neural networks are another pillar of deep learning that are employed for sequence modelling problems. They are intended to process time sequences in remote sensing. RNN is a special type of neural network that can process $\{x_1, x_2, x_3 \ldots x_n\}$ sequence of values. It allows the processing of the input of variable length. The core element of RNN is the RNN unit. It comprises of input X and output Y and in between recurrent unit that represents the recurrence through self loop. RNN have internal state that is also called as the internal memory which is updated when the sequence is processed. The sequence of vector X is processed by applying the recurrence formula at every time step as shown in equation number 5 and 6. Next hidden state is dependent on the current input X_t and the previous hidden state h_{t-1}. The commonly used activation for RNN hidden state is tanh.

$$h(t) = f_{UW(X_t, h_{t-1})} \tag{5}$$

where U and W are weight metrics that are multiplied with X_t and h_{t-1}

$$Y(t) = f(V, h(t)) \tag{6}$$

where V is the weight matrix that is multiplied with hidden state that comes out of the RNN unit and is passed through activation function to provide the final outcome as—$Y(t)$.

RNN and its variants like Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU) are commonly adapted for remote sensing tasks. For instance, Lichao Mou [13] proposed a novel RNN model and applied it for hyper spectral image classification. The results show that method outperforms the classic CNN models. Josh David [14] used optical data and SAR data and applied RNN, LSTM and GRU for crop mapping.

2.2 Deep Networks for Generative Learning

Supervised learning is considered as the dominant paradigm in deep learning. However, unsupervised deep learning is seeking more and more attention of the researchers in the remote sensing community. The learning approach in which labelled data is not available is referred to as Generative or Unsupervised learning. Models try to infer the underlying relationship using this unlabeled training data. Generative adversarial networks (GAN) [15] and Auto encoder [16] are commonly used for unsupervised learning task.

Autoencoder

Auto encoders utilize neural network for the task of representation learning. It comprises of the input layer, one or more hidden layer and the output layer. First, the input layers encodes the information. The encoded information is stored in the hidden layer which is also called as the bottleneck layer. Later the decoder decodes the information. The encoder finds the compressed latent representation of input data and the Decoder decodes this compressed representation and provides the reconstruction of input $x, \hat{x}$ [17]. The network tries to minimize the error between the actual input x and the reconstructed input $\hat{x}$. This error is referred as the reconstruction error. The network minimizes the reconstruction error though backpropogation. The variants of the auto encoder are Sparse Autoencoder (SAE), Denoising Autoencoder (DAE), Contractive Auto encoder (CAE) and Variational Autoencoder (VAE) [18]. Autoencoders are typically used in remote sensing for image denoising, image compression and image generation. For instance, Xu [19] proposed an end-to-end SAR image compression convolutional neural network (CNN) model based on a variational autoencoder. Song [18] utilized the adversarial autoencoder for SAR image generation.

Generative Adversarial Network

Generative Adversarial Network (GAN) can learn interpretation from remote sensing dataset in an unsupervised manner. GAN generates the new data instances that resemble the data instances in training data. GAN comprises of two network models, such as the Generator model and the discriminator model that compete with each other. The generator network generates the fake images and tries to fool the discriminator. On the other hand, the discriminator network distinguishes between the fake generated image and real image [20]. Over the years variants of GAN as progressive GAN, Conditional GAN, Image-to-Image translation GANs, CycleGANs, Text-to-image GANs architecture have been proposed. Among them, Conditional GAN (CGAN) [21] and cycle-GAN [22] are commonly used for remote sensing data fusion [23]. Zhao [24] used GAN-based SAR-to-Optical image translation. Apart from the above mentioned unsupervised learning methods, Self-Organizing Map (SOM), Restricted Botshaman Machine (RBM) and Deep Belief Network (DBN) are also employed for remote sensing tasks.

2.3 Hybrid Deep Networks for Generative Learning

Supervised and unsupervised learning approach is generally used for various applications. Each learning approach has its own pros and cons. There exists another learning technique as Hybrid learning that takes advantages of the supervised and unsupervised approach to solve a particular task. The aim of the Hybrid deep networks is to use an integrated approach. For instance, Zang [25] used the convolutional denoising autoencoder (C-DAE) to reconstruct the speckle-free SAR images. Yuanyuan Zhou [26] used Deep Multi-Scale Recurrent Network that includes a unit for SAR image despeckling.

3 Recent Research Progress in Deep Learning in the Field of Remote Sensing

In this Section a detailed review of state of the art deep learning methods that are employed to optical, SAR and SAR-optical fusion-based satellite data from different perspectives as Image scene classification, Object detection, Semantic segmentation and Image Despeckling are presented. Table 1 represents the comparative analysis of various deep learning methods used in literature for the aforementioned tasks. These tasks are discussed in the following section.

3.1 Image/Scene Classification

Image classification is a way that classifies pixels in satellite image into one of the classes. For instance, Land use Land cover classification method classifies the pixels into one of the land cover classes as bare land, water bodies, built up area, forest, road, grassland and rock based on the spectral reflectance. Scene classification is similar to image classification however scene is much larger in contrast to normal image patch [39]. The image classification method includes the conventional remote sensing methods that use spectral indices methods for optical data, backscatter intensity for SAR data, statistical methods, machine learning methods and recent deep learning methods. Some of the notable deep learning methods used on remote sensing data are reviewed likewise. The methods like Convolution neural networks, auto encoders, sparse auto encoder, stacked auto encoder, Generative adversarial network (GAN) are progressing in optical and SAR remote sensing image scene classification [40]. In the literature, modern deep learning CNN architectures are implemented for optical and SAR imagery analysis utilizing different approaches like training CNNs from scratch, using pretrained CNNs as feature extractors and fine-tuning pre trained CNN. For instance, Shyamal [41] used UNet architecture for sugarcane crop classification from Sentinel 2 satellite imagery. Yang-Lang Chang

Table 1 Comparative analysis of deep learning methods applied on remote sensing tasks-Image classification, Object detection, Semantic segmentation and sar image despeckling

References	Task	Method	Dataset	Evaluation Index	Limitations
Anas Hasni [11]	Image classification	VGG 16	Moving and Stationary Target Acquisition and recognition (MSTAR)	Accuracy: 97.91%	Deeper architecture than VGG16 can be explored to improve accuracy
Yang-Lang Chang [27]	Image classification	Consolidated Convolutional Neural Network, C-CNN AUG	Indian Pines (IP), Pavia University (PU), and Salinas Scene (SA)	Accuracy: 99.43%	Impact of variation of datasize on proposed model is not explored
Shi and Zhang [28]	Image classification	Self-Compensating Convolution Neural Network	UCM21, RSSCN7, AID, NWPU-RESISC45, WHU-RS19, SIRI-WHU	Accuracy: 99.21%	More effective method of feature extraction can be explored and accuracy can be improved
Liu and Zhang [29]	Image classification	Multidimensional CNN Combined with an Attention Mechanism Mode	Salinas (SA), WHU-Hi-HanChuan (WHU), and Pavia University (PU) datasets	Accuracy: 96.71%	Accuracy can be improved
Nataliia Kussul [30]	Image classification	Deep Recurrent Neural Network LSTM for Crop Classification Task	Sentinel-1 and Sentinel-2 Imagery	Accuracy: 97.5%	Generalization is required
Jiankun Chen [31]	Semantic segmentation	Complex Valued Deep Semantic Segmentation Network	Sentinel-1 and Sentinel-2 Imagery	Accuracy: 94.89%	Usage of proposed method on SAR data
Yanjuan Liu [32]	Semantic segmentation	Deep transfer learning	Sentinel-1 and Sentinel-2 Imagery	mIoU: 88.21	Performance can be improved
Tuan Pham [33]	Semantic segmentation	Pyramid scene parsing network (PSPNet) for Semantic Road Segmentation	Cityscapes Dataset	mIoU: 64.75	Experiments can be done with other datasets
Chaojun Shi [34]	Semantic segmentation	CloudU-Net for cloud segmentation	GDNCI data set	mIoU: 0.927	Other datasets can be tested

(continued)

Table 1 (continued)

References	Task	Method	Dataset	Evaluation Index	Limitations
Morales et al. [35]	Semantic segmentation	Convolutional Neural Network (CNN) based on the Deeplab v3+ architecture	CloudPeru2	Accuracy: 97.50%	
Haohao Ren [36]	Object detection	Active self-paced deep learning (ASPDL)	Moving and stationary target acquisition and recognition (MSTAR)	Accuracy: 94.00%	Accuracy can be improved
Dong Li [37]	Object detection	Multidimensional domain deep learning network for SAR ship detection	SAR ship detection data set (SSDD)	mAP: 0.96	Methods can be evaluated with other datasets
Zhang [25]	SAR image despeckling	Convolutioal denoising autoencoder (CDAE)	SAR images	PSNR: 40.83	Generalization of method is required
Malsha [38]	SAR image despeckling	Transformer-Based encoder	Set12 dataset	PSNR: 24.56	Method can be tested with more real data

[27] combined 3D CNN and 2D CNN and proposed the Consolidated Convolutional Neural Network (C-CNN).The proposed method is evaluated on benchmark optical datasets and results have proved that the proposed method provides better Hyperspectral Image Classification in contrast to previous state of the art methods. Cuiping Shi [28] constructed a lightweight self-compensated convolution by reducing the number of filters. The authors have proposed lightweight modular self compensating convolution neural network (SCCNN) for remote sensing scene image classification based on self-compensated convolution and self-compensating bottleneck module (SCBM). The model can efficiently classify remote sensing optical images with less number of parameters and good accuracy as compared to state of art classification methods.

Small samples in hyper spectral images affect the performance of image classification with convolution neural networks. To address this problem Jinxiang Liu [29] used convolutional block self-attention module (CBSM) with 2D convolution neural network layer to achieve better performance for hyperspectral image classification. Further Anil Raj used One-shot learning-based deep learning model and utilized it for SAR ship classification. Hyperspectral images usually have high dimensions which affect the performance of image classification. To address this issue, Hüseyin Fırat [42] used a combination of 3D CNN, 2D CNN and depthwise separable convolution and further combined it with dimensionality reduction methods to improve classification performance of remote sensing images. Deep learning methods have shown significant results on optical data. Few researchers have explored synthetic aperture image scene classification with deep neural networks. Anas Hasni [11] used transfer learning approach that uses VGG16 pretrained convolution neural network as a feature extraction for SAR image classification. Xie et al. [43] employed stacked auto encoder (SAE) that can automatically extract features from polSAR data. The features are subsequently fed to softmax. The remarkable results show that deep learning methods learn feature representations and are effectively used for terrain surface classification using PolSAR images. Geng [44] proposed the deep convolutional autoencoder (DCAE) method that automatically extracts the features and perform classification with good accuracy. Most of the methods observed in literature are based on single image classification. Teimouri [45] used a different approach of time series of SAR for crop classification. Further Nataliia Kussul [30] used SAR Optical data acquired from Sentinel-1 and Sentinel-2 and implemented recurrent neural network for crop mapping. Alessandro Lapini [46] implemented the deep convolution neural network on optical and SAR data for agricultural area classification. The challenges in the said task include large variance of scene scale, low between class separability and coexistence of multiple ground objects [40].

Hence, it is apparent that deep learning methods for Optical, SAR and PolSAR data have advanced considerably in past few years. In spite of these progresses, it has observed that there are many fields that could be explored to achieve lightweight deep learning models pertaining to remote sensing satellite data.

3.2 Semantic Segmentation

Semantic segmentation refers to the task of grouping the objects related to same class in satellite images. Over the years several deep learning methods have shown superior performance in semantic segmentation task. Modern deep learning architectures like AlexNet, VGGNet, GoogleLeNet are the base for semantic segmentation. Shelhamer [47] developed a fully convolutional network by extending AlexNet, VGGNet and GoogLeNet. Various methods like Fully Convolutional Network, U-Net (2015), SegNet (2017), DeepLab (2018) and its variants like DeepLab V3, DeepLab V3+ are commonly adapted for semantic segmentation task. Computational complexity is the major concern for remote sensing image segmentation. To deal with this researchers have utilized Densely Connected Convolutional Network (DenseNet) [48], ShuffleNet [49] for building semantics segmentation architecture for remote sensing data.

Semantic segmentation is implemented for various applications. For instance, Tuan Pham [33] used the Fully Convolutional Network**,** Pyramid scene parsing network (PSPNet) and SegNet for Semantic Road Segmentation using Deep Learning. Other than satellite image complexity, cloud cover is another hurdle in optical remote sensing data. The cloud cover limits the usage of such optical data for various applications. Giorgio Morales [35] proposed an efficient method that performs cloud segmentation in multispectral satellite images using a Convolutional

Neural Network. Further, Chaojun Shi [34] utilized dilated convolution to build on the convolutional neural network named CloudUNet for cloud segmentation. SAR image is complex valued particularly for multichannel coherent images like polarimetric SAR. Jiankun Chen [31] attempts to deal with this issue by proposing an improved semantic segmentation network named complex valued SegNet (CVSegNet). Deep learning algorithm perform well when huge amount of data is available for training of the model. Practically, it is difficult to get the labelled data for remote sensing. This poses certain limitations on usage of deep neural networks in remote sensing. Yanjuan Liu [32] proposed the deep transfer learning method that transfers improved Deeplabv3+ from SAR imagery to SAR and optical fusion imagery.

3.3 Object Detection

Object detection is the method of detecting instances of objects of a particular class of interest within the satellite image and locate the position of those objects in the satellite image. Object detection in optical remote sensing has been used in wide range of applications for precision agriculture, Landuse Landcover mapping, hazard detection etc. In recent years deep learning methods are extensively used for object detection. It has shown promising results on optical and SAR data. Here, Deep learning research progress pertaining to optical data followed by SAR is reviewed. Remarkable progress in deep learning has created a new pathway for large object detection. The variants of convolutional neural networks like region-based convolution neural networks (R-CNN) and faster region-based convolutional neural networks (Faster R-CNN),Single shot detector (SSD),You only look once (YOLO) are widely used and show promising results on large object detection. However, they are not able to detect small objects. Several efforts are taken to address this issue. Yun Ren [50] used modified Faster R-CNN small object detection in optical remote sensing images. The object detection is further extended to SAR images. Haohao Ren [36] proposed and implemented an active self-paced deep learning (ASPDL) method on SAR data for automatic target recognition. The experimental results have shown that ASPDL outperform state of art algorithms. In spite of initial success, performance of object detection is affected due to rotations, and the complex background. Dong Li [37] tackled this issue by proposing the Multidimensional Domain Deep Learning Network for SAR Ship Detection. In short, object detection from optical and SAR images can be achieved using deep learning methods.

3.4 SAR Image Despeckling

SAR data is highly contaminated due to speckle noise. This is a challenge in SAR image interpretation. Researchers have explored various supervised and unsupervised

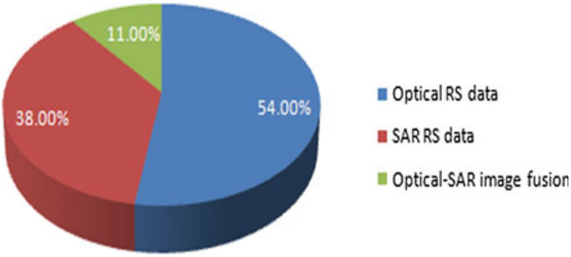


Fig. 2 Analysis of usage of deep learning models on Remote sensing data

methods of deep learning for SAR image despeckling. For instance, the simple convolution network is widely used for image despeckling, however, the texture of image is lost. Mohanakrishnan [51] proposed the Modified Convolutional Neural Network (M-CNN) algorithm that uses dilated convolution as convolution and Leaky ReLU as transfer function. The proposed model provides promising results for SAR despeckling. Shujun Liu [38, 52] used Multi-Weighted Sparse Coding for despeckling. Further Perera [38], Zhang [25] proposed the Transformer-Based and convolutional denoising autoencoder (C-DAE) model for SAR image despeckling.

Based on literature review, Fig. 2 depicts analysis of usage of deep learning models for the above-mentioned tasks pertaining to optical data, SAR data and Optical-SAR image fusion data.

4 Remote Sensing Benchmark Datasets

Deep learning is widely used for remote sensing applications. Deep learning models need good quality datasets as quality and quantity of labelled training data directly impacts the performance of the deep learning model. Getting good quality dataset is of prime importance. Several researchers have published the datasets for remote sensing data that include optical data, Synthetic aperture radar data (SAR) and optical-SAR fusion based approach. Figure 3 depicts the datasets that are commonly used to test the performance of novel deep learning methods and another category used by the researchers to solve a particular use case in remote sensing task are considered.

5 Evaluation Criteria

The evaluation matrix is a way to evaluate the performance of the deep learning model. The common evaluation metrics for classification include Accuracy, Precision, Sensitivity, Specificity, F1-Score, False Positive Rate (FPR), Area Under the ROC Curve and Kappa. The following are the considerations: Correctly classified instances are represented as True Positive (TP) whereas the instances which are negative and predicted as negative are represented as True Negative (TN). The instances

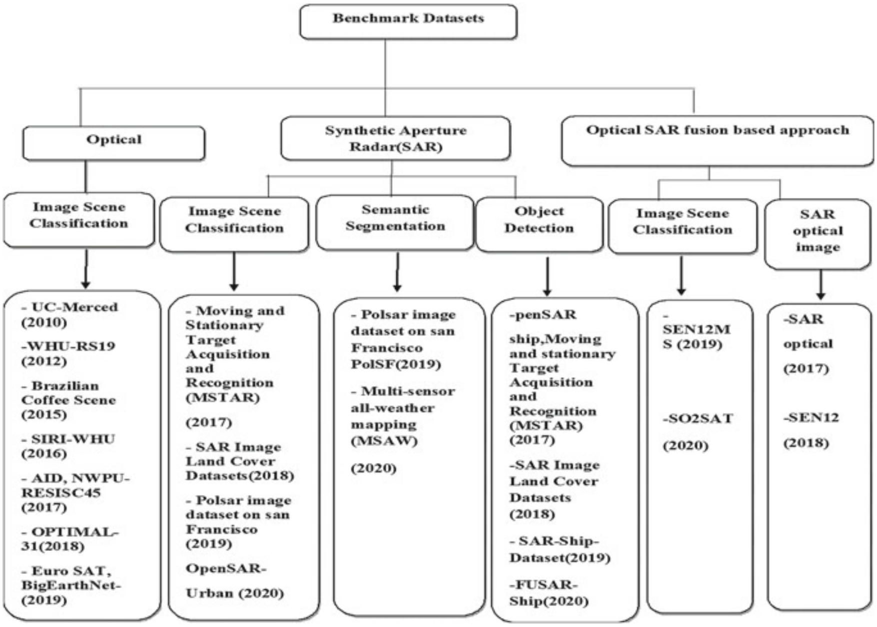

Benchmark Datasets
Optical
Synthetic Aperture Radar(SAR)
Optical SAR fusion based approach
Image Scene Classification
Image Scene Classification
Semantic Segmentation
Object Detection
Image Scene Classification
SAR optical image
- UC-Merced (2010)
-WHU-RS19 (2012)
- Brazilian Coffee Scene (2015)
- SIRI-WHU (2016)
- AID, NWPU-RESISC45 (2017)
- OPTIMAL-31(2018)
- Euro SAT, BigEarthNet-(2019)
- Moving and Stationary Target Acquisition and Recognition (MSTAR)
(2017)
- SAR Image Land Cover Datasets(2018)
- Polsar image dataset on san Francisco (2019)
OpenSAR-
Urban (2020)
- Polsar image dataset on san Francisco PolSF(2019)
- Multi-sensor all-weather mapping (MSAW)
(2020)
-penSAR
ship,Moving and stationary Target Acquisition and Recognition (MSTAR) (2017)
-SAR Image Land Cover Datasets (2018)
- SAR-Ship-Dataset(2019)
-FUSAR-Ship(2020)
-
SEN12MS (2019)
-SO2SAT
(2020)
-SAR
optical
(2017)
-SEN12
(2018)

Fig. 3 Benchmark datasets [39, 40, 53]

which are predicted as positive but are actually negative are represented as False positive (FP) as well the instances which are predicted as negative but are actually positive and are represented as False Negative (FN). Table 2 represents the evaluation metrics used in literature for various remote sensing tasks.

6 Conclusion and Future Directions

In this article, a brief review of the state of the art deep learning methods for optical and synthetic aperture radar (SAR) satellite imagery has been considered. The discussion is based on relevant deep learning architectures that are used for solving remote sensing tasks. Further research progress of deep learning is analyzed with reference to image classification, object detection, semantic segmentation and SAR image despeckling. Remote sensing benchmark datasets and evaluation criteria are further discussed. The potential of deep learning methods for the underexplored area of synthetic aperture radar, SAR-Optical data fusion have been highlighted lastly. The promising research directions for deep learning and remote sensing are presented in the section below:

1. Optical-SAR image fusion: Most of the work in remote sensing with deep learning is limited by optical data. However, SAR is not much explored. There

Table 2 Evaluation criteria

Remote sensing task used in literature	Evaluation metrics	Mathematical representation
Classification [11, 27–29, 30]	Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
	Precision	$Precision = \frac{TP}{TP+FP}$
	Recall/Sensitivity	$Recall = \frac{TP}{TP+FN}$
	Specificity	$Specfcity = \frac{TN}{FP+TN}$
	F1-Score	$F1score = 2X \frac{PrecisionXRecall}{Precision+Recall}$
	Kappa coefficient	$(k) = \frac{Po-Pe}{1-Pe}$
Semantic segmentation [31–35]	IoU or Jaccard Index	$\mathrm{IoU} = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
	Dice	$Dice(A, B) = \frac{2\lvert A \cap B \rvert}{\lvert A \rvert + \lvert B \rvert}$
Object detection [54]	Average Precision	$AP = \int_0^1 p(t)dt$
	mean Average Precision (mAP)	$mAP = \frac{1}{N} \sum_{i=1}^{N} APi$

are lots of research opportunities for SAR data and fusion based approach of SAR and Optical data. Solving remote sensing problems with an integrated approach may accelerate the research in this area.

2. Inadequate datasets: The survey shows that limited public datasets are available for SAR and SAR-optical fusion. Building novel dataset for SAR modalities can unlock limitations of SAR. Building dataset by utilizing deep learning methods could benefit another research direction.
3. Building up of novel deep learning models: State of the deep learning models are successfully used for remote sensing task. However, the usage of these models on mobile devices is challenging due to limited power and memory. Building lightweight deep learning architecture for next generation mobile devices could be a novel contribution.
4. Image scene understanding: SAR images are complex in nature. It is difficult to understand these images without the help of remote sensing experts. Remote sensing image scene understanding with deep learning can be another future direction.

Concluding, this review article provides pathway to the budding researchers who wish to work in the domain of remote sensing and deep learning.

References

1. Pagot E (2008) Systematic study of the urban postconflict change classification performance using spectral and structural features in a support vector machine. IEEE J Select Topics Appl Earth Observ Remote Sens 1:120–128. https://doi.org/10.1109/JSTARS.2008.2001154
2. Feng W (2017) Random forest change detection method for high-resolution remote sensing images. J Surv Mapp 46(11):90–100
3. Li D (2014) Automatic analysis and mining of remote sensing big data. Acta Geodaetica et Cartographica Sinica 43(12):1211–1216
4. Li G, Jiajun L (2020) Automatic analysis and intelligent information extraction of remote sensing big data. J Phys: Conf Series 1616. 012003. https://doi.org/10.1088/1742-6596/1616/1/012003
5. Zhu M (2019) A review of researches on deep learning in remote sensing application. J Geosci 10:1–11. https://doi.org/10.4236/iig.2019.101001
6. Gong JY, Ji SP (2017) From photogrammetry to computer vision. Geomat Inf Sci Wuhan Univ 42:1518–1522
7. Jiyana Gong S (2018) Photogrammetry and Deep Learning. Acta Geodaetica et Cartographica Sinica 47:693–704
8. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classifcation with deep convolutional neural networks. Commun ACM 60(6):84–90
9. Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning, vol 1. MIT press, Cambridge
10. Yamashita R, Nishio M, Do RKG et al (2018) Convolutional neural networks: an overview and application in radiology. Insights Imag 9:611–629. https://doi.org/10.1007/s13244-018-0639-9
11. Hasni A, Hanifi M, Anibou C (2020) Deep Learning for SAR image classification. Intell Syst Appl. https://doi.org/10.1007/978-3-030-29516-5_67
12. Makantasis K, Karantzalos K, Doulamis A, Doulamis N (2015) Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: Proceedings of IEEE International Geoscience Remote Sensing Symposium, Milan, Italy, pp 4959–4962
13. Mou L, Ghamisi P, Zhu XX (2017) Deep recurrent neural networks for hyperspectral image classification. IEEE Trans Geosci Remote Sens 55(7):3639–3655. https://doi.org/10.1109/TGRS.2016.2636241
14. Bermúdez JD et al (2017) Evaluation of recurrent neural networks for crop recognition from multitemporal remote sensing images. In: Anais do XXVII Congresso Brasileiro de Cartografia
15. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets in Advances in neural information processing systems 2014:2672–2680
16. Kingma P, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
17. Zhu XX, Montazeri S, Ali M, Hua Y, Wang Y, Mou L, Shi Y, Xu F, Bamler R (2020) Deep learning meets SAR. arXiv preprint arXiv:2006.10027
18. Song Q, Xu F, Zhu XX, Jin YQ (2022) Learning to generate SAR images with adversarial autoencoder. IEEE Trans Geosci Remote Sens 60:1–15. Art no. 5210015. https://doi.org/10.1109/TGRS.2021.3086817
19. Xu Q et al (2022) Synthetic aperture radar image compression based on a variational autoencoder. IEEE Geosci Remote Sens Lett 19:1–5. Art no. 4015905. https://doi.org/10.1109/LGRS.2021.3097154
20. Ben Hamida A, Benoit A, Lambert P, Ben Amar C (2018) Generative Adversarial Network (GAN) for remote sensing images unsupervised learning. In: RFIAP 2018, AFRIF, SFPT, IEEE GRSS, Jun 2018, Marne-la-Vallée, France. ffhal-0197031
21. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784

22. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232
23. Liu P (2021) A review on remote sensing data fusion with generative adversarial networks (GAN). TechRxiv.Preprint
24. Zhao Y, Celik T, Liu N, Li H-C (2022) A comparative analysis of GAN-based methods for SAR-to-optical image translation. IEEE Geosci Remote Sens Lett. https://doi.org/10.1109/LGRS.2022.3177001
25. Qianqian Z, Sun R. SAR image despeckling based on convolutional denoising autoencoder. https://doi.org/10.13140/RG.2.2.24936.29443
26. Zhou Y, Shi J, Yang X, Wang C, Kumar D, Wei S, Zhang X (2019) Deep multi-scale recurrent network for synthetic aperture radar images despeckling. Remote Sens 11(21):2462. https://doi.org/10.3390/rs11212462
27. Chang Y-L, Tan T-H, Lee W-H, Chang L, Chen Y-N, Fan K-C, Alkhaleefah M (2022) Consolidated convolutional neural network for hyperspectral image classification. Remote Sens 14:1571. https://doi.org/10.3390/rs14071571
28. Shi C, Zhang X, Sun J, Wang L (2022) Remote sensing scene image classification based on self-compensating convolution neural network. Remote Sens 14:545. https://doi.org/10.3390/rs14030545
29. Liu J, Zhang K, Wu S, Shi H, Zhao Y, Sun Y, Zhuang H, Fu E (2022) An investigation of a multidimensional CNN combined with an attention mechanism model to resolve small-sample problems in hyperspectral image classification. Remote Sens 14:785. https://doi.org/10.3390/rs14030785
30. Kussul N, Lavreniuk M, Shumilo L (2020) Deep recurrent neural network for crop classification task based on Sentinel-1 and Sentinel-2 imagery. In: IGARSS 2020—2020 IEEE international geoscience and remote sensing symposium, pp 6914–6917. https://doi.org/10.1109/IGARSS39084.2020.9324699
31. Chen J, Qiu X (2019) Equivalent complex valued deep semantic segmentation network for SAR images. In: International applied computational electromagnetics society symposium—China (ACES), pp 1–2.https://doi.org/10.23919/ACES48530.2019.9060476
32. Liu Y, Kong Y (2021) A novel deep transfer learning method for SAR and optical fusion imagery semantic segmentation. In: IEEE international geoscience and remote sensing symposium IGARSS, pp 4059–4062.https://doi.org/10.1109/IGARSS47720.2021.9553751
33. Pham T (2020) Semantic road segmentation using deep learning. Applying New Technology in Green Buildings (ATiGB) 2021:45–48. https://doi.org/10.1109/ATiGB50996.2021.9423307
34. Shi C, Zhou Y, Qiu B, Guo D, Li M (2021) CloudU-Net: a deep convolutional neural network architecture for daytime and nighttime cloud images' segmentation. IEEE Geosci Remote Sens Lett 18(10):1688–1692. https://doi.org/10.1109/LGRS.2020.3009227
35. Morales G, Ramírez A, Telles J (2019):End-to-end cloud segmentation in high-resolution multispectral satellite imagery using deep learning. In: IEEE XXVI international conference on electronics, electrical engineering and computing (INTERCON), pp 1–4. https://doi.org/10.1109/INTERCON.2019.8853549
36. Ren H, Yu X, Bruzzone L, Zhang Y, Zou L, Wang X (2022) A Bayesian approach to active self-paced deep learning for SAR automatic target recognition. IEEE Geosci Remote Sens Lett 19:1–5. Art no. 4005705. https://doi.org/10.1109/LGRS.2020.3036585
37. Li D, Liang Q, Liu H, Liu Q, Liu H, Liao G (2022) A novel multidimensional domain deep learning network for SAR ship detection. IEEE Trans Geosci Remote Sens 60:1–13. Art no. 5203213. https://doi.org/10.1109/TGRS.2021.3062038
38. Parera MV. Transformer based SAR image despeckling.arXiv:2201.09355
39. Zhu XX et al (2021) Deep learning meets SAR: concepts, models, pitfalls, and perspectives. IEEE Geosci Remote Sens Mag 9(4):143–172. https://doi.org/10.1109/MGRS.2020.3046356
40. Cheng G, Xie X, Han J, Guo L, Xia G-S (2020) Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities. IEEE J Select Topics Appl Earth Observ Remote Sens 13:3735–3756. https://doi.org/10.1109/JSTARS.2020.3005403

41. Virnodkar S, Pachghare VK, Murade S (2021) A technique to classify sugarcane crop from Sentinel-2 satellite imagery using U-Net architecture. In: Progress in advanced computing and intelligent engineering. Advances in Intelligent Systems and Computing, vol 1199. Springer, Singapore. https://doi.org/10.1007/978-981-15-6353-9_29
42. Fırat H, Emin Asker M, Hanbay D (2022) Classification of hyperspectral remote sensing images using different dimension reduction methods with 3D/2D CNN, Remote Sens Appl: Soc Environ 25:100694.ISSN 2352-9385.https://doi.org/10.1016/j.rsase.2022.100694
43. Xie H, Wang S, Liu K, Lin S, Hou B (2014):Multilayer feature learning for polarimetric synthetic radar data classification. In: IEEE international geoscience and remote sensing symposium (IGARSS)
44. Geng J, Fan J, Wang H, Ma X, Li B, Chen F (2015):High-resolution SAR image classification via deep convolutional autoencoder. IEEE Geosci Remote Sens Lett 12(11): 2351–2355
45. Teimouri N, Dyrmann M, Jørgensen RN (2019) A novel spatiotemporal FCN-LSTM network for recognizing various crop types using multi-temporal radar images. Remote Sens 11(8):990
46. Lapini A et al (2020) Application of deep learning to optical and SAR images for the classification of agricultural areas in Italy, In IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 2020, pp 4163–4166. https://doi.org/10.1109/IGARSS39084.2020.9323190
47. Shelhamer E, Jonathan L, Trevor D (2017) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651
48. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
49. Zhang X, Zhou X, Lin M, Sun J (2018):ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: The IEEE conference on computer vision and pattern recognition (CVPR)
50. Ren YZ, Changren X, Shunping (2018) Small object detection in optical remote sensing images via modified faster R-CNN. Appl Sci 8:813, 2076–3417. https://doi.org/10.3390/app8050813
51. Mohanakrishnan P, Suthendran K, Pradeep A, Yamini AP (2022) Synthetic aperture radar image despeckling based on modifed convolution neural network. Appl Geomatics. https://doi.org/10.1007/s12518-022-00420-8
52. Liu S, Pu N, Cao J, Zhang K (2022) Synthetic aperture Radar image despeckling based on multi-weighted sparse coding. Entropy 24:96. https://doi.org/10.3390/e24010096
53. Schmitt M, Hughes LH, Zhu XX (2018) The Sen1–2 dataset for deep learning In Sar-optical data fusion. ISPRS Ann Photogramm Remote Sens Spatial Inf Sci IV-1:141–146. https://doi.org/10.5194/isprs-annals-IV-1-141-2018
54. Cheng G, Han J (2016) A survey on object detection in optical remote sensing images. ISPRS J Photogramm Remote Sens 117:11–28.ISSN 0924-2716. https://doi.org/10.1016/j.isprsjprs.2016.03.014

TSK-Based Type-2 Fuzzy Analysis of Infrared Spectroscopic Data for Classification of Touch-Induced Affection

Mousumi Laha, Dipdisha Bose, and Amit Konar

Abstract The paper introduces a novel approach to categorize the hemodynamic response of subjects due to arousal of touch induced affection classes such as Respect, Love, Fondness and Devotion using a TSK-based Type-2 Fuzzy classifier. The main contribution of the paper is to design the novel TSK-based Interval Type-2 Fuzzy classifier to classify the finer changes in affective emotions using the hemodynamic response of a subject, when she comes in contact with her mother, spouse, child and also conveys her prayer to a model/sculpture of God by holding it with her palms. Experiments undertaken reveal that the brain activation patterns varies in different sub-regions over distinct time-windows for individual emotions. Relative performance analysis and statistical validation confirm the superiority of the proposed TSK-based Interval Type-2 Fuzzy classifier. Moreover, the proposed scheme has successfully been applied for assessing subjective sensitivity of healthy as well as psychiatric disordered people.

Keywords Touch-induced affection · Functional near infrared spectroscopy (f-NIRs) · Hemodynamic response · Interval Type-2 fuzzy classifier

1 Introduction

Since the beginning of the human civilization, 'touch' is adopted as a fundamental modality of nourishment specially for children and people suffering from psychological distresses (including stress, anxiety and depression) [1, 2]. However, the role of 'soft touch' in inducing affection is emphasized very recently in scientific publications [3, 4]. The true understanding of touch perception from the points of view of changes in activation of different brain regions remained a virgin area of research

M. Laha (✉) · A. Konar
Electronics and Telecommunication Engineering Department, Jadavpur University, Kolkatta, India
e-mail: lahamou@gmail.com

D. Bose
School of Bioscience and Enginnering, Jadavpur University, Kolkatta, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_12

in brain and cognitive sciences till this date. This paper explores the possibility of inducing 4 different classes of affective emotions by touching one subject's palm by her family members (spouse, child and parents). It makes an attempt to assess the true 'nature of affection' (love, fondness, respect and devotion) aroused in a subject during the period of contact of her palm with her individual family members as well as a sculpture of God/Goddess she is habituated to worship, directly from the hemodynamic response of her brain.

Similar works have been undertaken recently [5] using EEG-based BCI. EEG offers the advantage to responding to instantaneous changes in the input stimuli. However, due to poor spatial resolution, it is unable to localize the brain activation regions precisely, and also fail to provide the accurate degree of activations at different brain regions because of volume conductivity of the scalp [6]. Functional Magnetic Resonance Imaging (f-MRI) [7] is a good choice to get rid of the above problems. However, because of excessive cost of the f-MRI device, most of the small BCI labs in the world cannot afford it. Rather, these labs utilize functional Near Infrared Spectroscopy (f-NIRs) [8] to determine the brain activations at different locations in the brain. In this paper, we would deal with f-NIRs device to capture the brain activations. These devices measure oxygenated and de-oxygenated blood concentrations, thus offering the degree of activation in a brain region based on the consumption of oxygen in the local tissues of the region. It is noteworthy that during arousal of an emotion, one or more activation regions in the brain are found active. The natural question that appears immediately: can we recognize the emotion of a person from his brain activation regions? This paper will ultimately give an answer to this important question. The approach adopted to handle the present problem is outlined next.

First, the brain regions responsible for a selected affective emotion are identified. It is important to learn that for the four classes of emotion chosen, the common brain regions are temporal and pre-frontal lobes. However, there exist temporal variations in the activation patterns within the sub-regions of an activated region. For instance, if the emotion refers to love, the affected brain regions show high activations first in the hippocampus (temporal lobe) and then shifts towards Orbito-frontal cortex (in the pre-frontal lobe). In case of parental respect, a high activation first appears in the Orbito-frontal cortex, which has a gradual shift towards the temporal region. These observations jointly reveal that the training instances to be developed should include temporal variation in the regions. After the emotions are aroused, we ask the subject about his/her feeling and thus fix up the emotion as the class and the temporal variations in the regions as the features to develop the training instances.

Any traditional classification algorithm could be employed to train the classifiers by the generated training instances. However, because of intra-subjective and inter-subjective variations, a fuzzy classifier is a better choice [9]. In this paper, a Takagi–Sugeno-Kang (TSK) [10, 11] based Interval Type-2 Fuzzy (IT2F) Classifier is employed to handle the present problem. Takagi–Sugeno-Kang (TSK) based model is advantageous to its competitor Mamdani based model with respect to structure of the fuzzy rules. Both the Mamdani and Takagi–Sugeno-Kang (TSK) model include similar antecedent part, but they differ in the consequent side [12]. While Mamdani based rules have a fuzzy quantified proposition of the output variable in the consequent, TSK-based model employs a linear function of the antecedent variables as the consequent. So, the variable in the consequent can be obtained in defuzzified form, and needs no additional defuzzification and type-reduction [13].

There is still a subtle problem that needs to be clarified at the beginning. How do we ensure that the emotions we capture from the subject are inherently accurate? For instance, touching wife's palm by her husband may not ensure transmission of love, in case the wife is aware of her husband's psychological involvement with a number of girl friends. Similarly, a touch by a mother to her daughter may not result in a glimpse of respect in the daughter, if the latter dislikes her mother. This took a lot of time to identify individuals for the experiments. The inter-personal relationships of the subject with spouse, parents and children were asked, and the subject was chosen after confirmation that she/he has a good relation with his/her family members. So, now we can ensure that a touch by a spouse at the palm of the subject may yield love, while a touch by parents results in a matter of respect in the subject, and so on.

The paper is divided into five sections. In Sect. 2, the principles and methodologies of the proposed scheme are illustrated using a proposed architecture of the TSK-based type-2 fuzzy classifier. Section 3 is concerned with the experiments and results. Experiments undertaken in this section reveal that the classification accuracy of the proposed TSK-based type-2 fuzzy classifiers yields better performance over the traditional classifiers. The performance analysis and the statistical validation of this novel approach are given in Sect. 4. Statistical test undertaken also confirms the superiority of the proposed technique over others. The conclusions arrived at the end of the paper are summarized in Sect. 5.

2 Principles and Methodology

This section gives a brief description of the principles and methodologies that have been undertaken to classify four distinct affection classes from the hemodynamic response of the subjects. The touch induced affective emotion classification is performed in five steps: (a) time-windowing and Data acquisition (b) normalization of the raw f-NIRs signals, (c) pre-processing and artifact removal, (d) feature extraction and selection from the filtered f-NIRs data and (e) classification. Figure 1 provides the basic block diagram of touch induced affective emotion classification using f-NIRs device.

2.1 Time-Windowing and Data Acquisition

In the present context, the f-NIRs data acquisition is carried out over various time-windows across trials. Each trial includes 4 distinct touch patterns for the arousal of four emotions such as Devotion, Respect, Love and Fondness. The subject arouses her affective emotions, when she comes in contact (due to touch) with her husband, child,

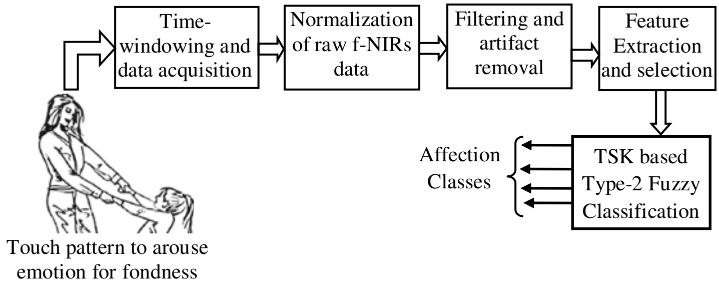

Time-windowing and data acquisition
Normalization of raw f-NIRs data
Filtering and artifact removal
Feature Extraction and selection
TSK based Type-2 Fuzzy Classification
Affection Classes
Touch pattern to arouse emotion for fondness

Fig. 1 Block diagram of the complete system

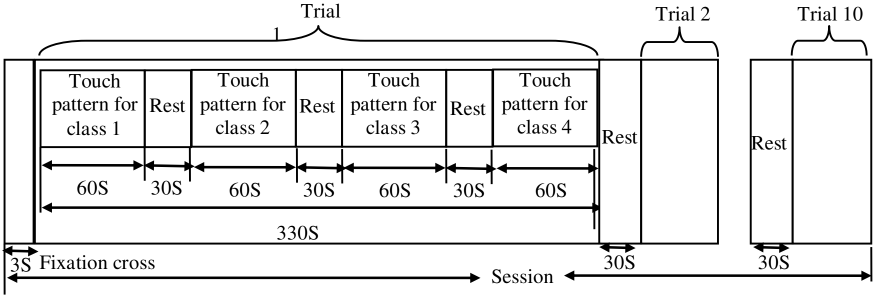

Trial 1
Trial 2
Trial 10
Touch pattern for class 1
Rest
Touch pattern for class 2
Rest
Touch pattern for class 3
Rest
Touch pattern for class 4
Rest
Rest
60S
30S
60S
30S
60S
30S
60S
330S
3S
Fixation cross
Session
30S
30S

Fig. 2 Presentation of touch-induced affection stimuli for a session

mother and model of the Almighty over distinct time-intervals. The hemodynamic response is acquired from the scalp of the subject for 60 s duration with a time interval (rest period) of 30 s. Consequently, the total duration of each trial is 330 s containing (60 s × 4) = 240 s for 4 distinct touch patterns and (30 s × 3)s = 90 s for 3 rest periods. The experiment includes 10 such trials in a session. Each session starts with 3 s fixation cross. To overcome the contamination effect between two successive trials, an interval of 30 s time gap is maintained over a session. Each session is repeated for 5 times in a day. Figure 2 provides one timing diagram of trials over a session.

2.2 Normalization of Raw f-NIRs Data

The following principle is adopted for the normalization of the hemodynamic response. Let $C_{HbO_\alpha}(t)$ be the oxygenated hemoglobin concentration of α-th channel at time t. Similarly, $C_{HbR_\alpha}(t)$ be the de-oxygenated hemoglobin concentration of α-th channel at time t. The normalization of $C_{HbO_\alpha}(t)$ and $C_{HbR_\alpha}(t)$ at a given channel are evaluated by the following 2 parameters:

$$^{\max}C_{HbO} = Max_t(C_{HbO_\alpha}(t) : t_0 \le t \le T, \forall\alpha) \tag{1}$$

$$^{\min}C_{HbR} = Min_t(C_{HbR_\alpha}(t) : t_0 \le t \le T, \forall\alpha) \tag{2}$$

where t_0 and T stand for the starting and the end time points of an experimental trial for a particular touch pattern of a specific subject [14], respectively.

The cerebral oxygen change in the temporo-prefrontal region is normalized in [0,1] by the following transformation.

$$d_\alpha(t) = \frac{(C_{HbO_\alpha}(t) - C_{HbR_\alpha}(t))}{^{\max}C_{HbO}(t) - {}^{\min}C_{HbR}(t)} \tag{3}$$

The sampling frequency of the particular device used in the experiment is 7.8 Hz. During the training phase, each touch pattern has 60 × 7.8 = 468 samples/s.

2.3 Artifact Removal from Normalized f-NIRs Data

Due to the non-stationery characteristics of brain signals, the acquired f-NIRs signals are not free from artifacts. To eliminate the artifacts from the raw f-NIRs signals, three individual steps are undertaken. In the first step, the *Common Average Referencing (CAR)* [15] has been performed to eliminate the motion artifacts.

Let $d_\alpha(t)$ be the normalized oxygen consumption of channel α at time *t* and $d_{avg}(t)$ be the average oxygen consumption over all channels(=20) at time *t*. Thus the *common average referenced* signal $CAR_\alpha(t)$ for channel α = 1 to 20 is evaluated by

$$CAR_\alpha(t) = d_\alpha(t) - d_{avg}(t). \tag{4}$$

In the second step, the $CAR_\alpha(t)$ signals are passed through the Elliptical band pass filter [16] of order 10, to eliminate the physiological artifacts like eye-blinking, heart rate, respiration etc. The pass band frequency of the Elliptical band pass filter is (0.1–8) Hz. Finally, the independent component analysis (ICA) [17] has been performed to determine the highest correlation between f-NIRs signals acquired from other channels.

2.4 Feature Extraction and Selection

To extract the important set of features from the filtered f-NIRs signals, the 60 s time interval for each touch pattern is divided into 6 equal time frames. From each time frame two sets of features (Such as static features and dynamic features) are extracted.

Static features include mean variance, skewness, kurtosis, average energy and the dynamic features include the changes in static features between two consecutive time frames [18].

In the present application, ($5 \times 6 =$) 30 static features and ($5 \times 5 =$) 25 dynamic features, altogether ($30 + 25 =$) 55 features are extracted for a given channel. Thereby, 20 channels yield $55 \times 20 = 1100$ features for each trial of a given touch pattern. Next, from 1100 features, 50 best features are selected using Evolutionary algorithm for classifier training. Here, the well-known Differential Evolution (DE) algorithm has been used for its simplicity, low computational overhead [19, 20]. Now, to classify 4 emotions, each session includes 10 trials and 5 such sessions are prepared for each touch pattern. Consequently, for 30 healthy subjects 30 × 5 sessions × 10 trials/session = 150 trials are generated for each touch pattern.

Finally, for 4 touch patterns $4 \times 150 = 600$ training instances are fed to the proposed TSK based Interval type-2 fuzzy classifier to classify 4 distinct emotions aroused from the hemodynamic response of a subject.

2.5 Proposed TSK-Based Interval Type-2 Fuzzy (TSK-IT2Fs) Classifier Design

A novel TSK based Interval Type-2 Fuzzy (IT2F) classifier model has been presented here to classify 4 affective emotions of a subject from their acquired hemodynamic responses.

Let, f_1, f_2, ..., f_n be n number of features extracted from the respective channel positions of the brain of a subject during the experiment. The experiment is performed over 5 sessions in a day, where each session comprises 10 experimental trials. Let, $f_{i,j}$ is $\tilde{A}_{i,j}$ be a fuzzy proposition used to build up the antecedent part of the fuzzy rule j. Now to construct $\tilde{A}_{i,j}$ both intra-session and inter-session variations have been considered.

Suppose, $f_{i,h,s,j}$ be the i-th feature extracted on session s in trial h of rule j. The mean and the variance of the feature i over a session s of that rule are respectively given by

$$\overline{f}_{i,s,j} = \left(\sum_{h=1}^{10} f_{i,h,s,j}\right)/10 \tag{5}$$

$$\sigma^2_{i,s,j} = \sum_{h=1}^{10} f_{i,s,h,j} - \overline{f}_{i,s,j})^2/10. \tag{6}$$

Now, a type-1 Gaussian MF $G(\overline{f}_{i,s,j}, \sigma^2_{i,s,j})$ is constructed to model the intra-session variation of the i-th feature extracted from rule j.

Now, the upper MF (UMF), of feature i of the j-th rule is considered as

$$UMF(f_{i,j}) = \overline{\mu}_{\tilde{A}_{i.j}}(f_{i,j}) = G_{i,j}(\overline{f}_{i,s,j}, \sigma^2_{i,s,j}) \tag{7}$$

where,

$$G(\overline{f}_{i,s,j}, \sigma^2_{i,s,j}) = \exp[-(f_{i,s,j} - \overline{f}_{i,s,j})^2 / 2\sigma^2_{i,s,j}] \tag{8}$$

Now, to construct the Lower MF (LMF) of $f_{i,j}$, we consider the concentration of the UMF.

Mathematically,

$$LMF(f_{i,j}) = Con(\overline{\mu}_{\tilde{A}_{i,j}}(f_{i,j})) = (\overline{\mu}_{\tilde{A}_{i,j}}(f_{i,j}))^2. \tag{9}$$

The TSK model proposed here employs type-2 fuzzy rules, where the *j*-th rule is given by.

$If\ f_1\ is\ \tilde{A}_{1,j},\ f_2\ is\ \tilde{A}_{2,j},\ ...,\ f_n\ is\ \tilde{A}_{n,j},\ Then\ y_j = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{a_{i,j}} * f_i + b_j$. Here, $f_1, f_2, ..., f_n$ together denotes a measurement point, and y_j denotes the signal power of the temporo-prefrontal region to classify effective emotion classes. The co-efficient $\mathrm{a_{i,j}}$ and b_j used in the classifier model are evaluated by classical least min-square technique [21, 22].

The proposed TSK-based IT2Fs model undertakes the following steps in order (Fig. 3).

1. Computations of Upper Firing Strength (UFS) and the Lower Firing Strength (LFS) for the *j*-th rule at the given measurement point are depicted by Eqs. (10) and (11) respectively.

$$UFS_j = \min[\overline{\mu}_{\tilde{A}_1}(f_1), \overline{\mu}_{\tilde{A}_2}(f_2), ..., \overline{\mu}_{\tilde{A}_n}(f_n)] \tag{10}$$

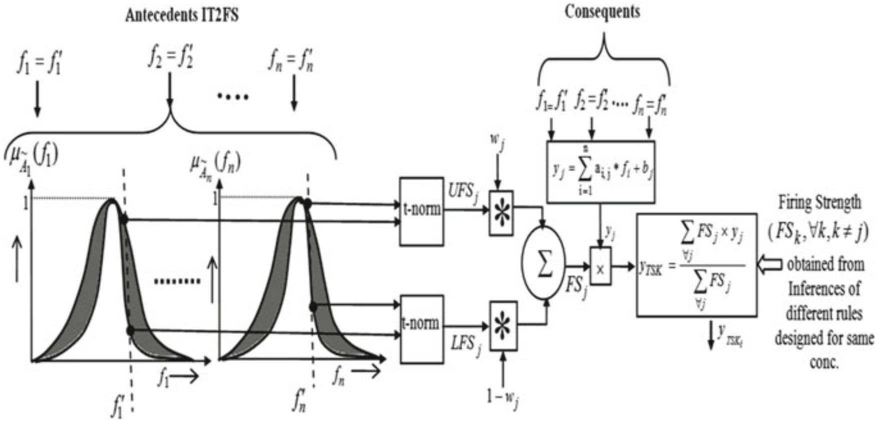


Fig. 3 Architecture of proposed TSK based interval type-2 fuzzy classifier

$$LFS_j = \min[\underline{\mu}_{\tilde{A}_1}(f_1), \underline{\mu}_{\tilde{A}_2}(f_2), ..., \underline{\mu}_{\tilde{A}_n}(f_n)] \tag{11}$$

2. Next, the firing strength (FS$_\text{j}$) of rule *j* is evaluated by taking the product of the weighted sum of UFS_j and LFS_j. The weights lie between [0, 1] hence, one weight is w_j and the other weight is $1 - w_j$. . Thus, the firiging Strength (*FS*) for rule *j* will be,

$$FS_j = w_j.UFS_j + (1 - w_j).LFS_j. \tag{12}$$

   Finally, Evolutionary algorithm (EA) has been utilized for optimal selection of the weights.
3. The resulting response of the type-2 TSK- based type-2 fuzzy classifier model is computed by

$$y_{TSK} = \frac{\sum\limits_{\forall j} FS_j \times y_j}{\sum\limits_{\forall j} FS_j}, \tag{13}$$

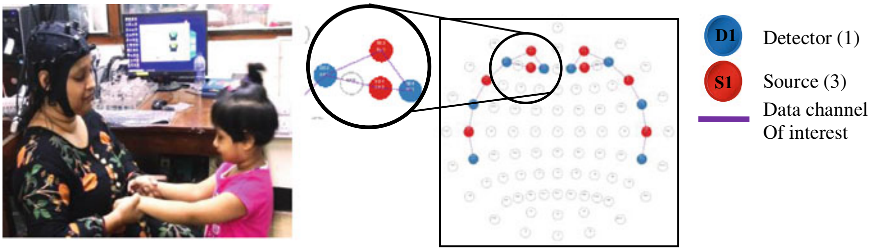
   where FS_j is the firing strength of the *j*-th rule.

Now, to classify four emotions classes from the measure of y_{TSK}, we divide the interval [0, $y_{TSK}^{\max}$] into 4 non-overlapped partitions, where each partition is segregated from its neighbors by two partition-boundaries. Thus for 4 partitions, we need to insert three partition boundaries. Let α_1 through α_3 be the three boundaries in [0, y_{TSK}], such that $y_{TSK}^{Max} > \alpha_3 > \alpha_2 > \alpha_1 > 0$.

Now, the boundaries α_1 through α_3 are evaluated by an Evolutionary algorithm. The motivation in the present context is to choose the parameters α_1 through α_3, so as to maximize the classification accuracy for a given set of training instances of affection classification.

3 Experiments and Results

Experiments are undertaken in 2 phases: Training phase and test phase. During the training phase, the weights: w_j and $(1 - w_j)$ for each rule *j* are tuned in order to maximize the classification accuracy for training instances of each class. After the training phase is over, we go for the test phase, where the affection-class of an unknown instance of brain response is provided as an input, and the class of affection is determined using the pre-trained IT2 fuzzy classifier.



(a) Experimental Set-up

(b) Source-detector connection of Temporo-prefrontal cortex 8×8 montage.

Fig. 4 **a** Experimental set-up. **b** Source-detector connection of Temporo-prefrontal cortex 8 × 8 montage

3.1 Experimental Framework and f-NIRs Data Acquisition

The experiment has been conducted in Artificial Intelligence laboratory of Jadavpur University, Kolkata, India [23]. The experimental setup is shown in Fig. 4a. A whole brain f-NIRs device, manufactured by NIRx Medical Technologies LLC, has been used to capture the hemodynamic response of the subject [24]. The f-NIRs device includes 8 Infrared sources and 8 Infrared detectors, which form $8 \times 8 = 64$ channels and placed over the scalp of the subject. Among 64 channels, 20 nearest neighboring source-detector pairs are utilized to execute the experiment (Fig. 4b). The experiment has been performed over ten healthy and normal volunteers (mostly women), in the age between 25 and 32 years, with her husband, mother and her own child. Each women volunteer is requested to arouse their emotions, when they come in physical contact with their husband, mother and their 2 to 4 year-old-children.

3.2 Experiment 1: (Automatic Feature Extraction to Discriminate 4 Affective Emotions)

The motivation of the present experiment is to discriminate the f-NIRs features for 4 affective emotions aroused by the subjects. Differential Evolutionary (DE) algorithm has been adopted to select the best possible f-NIRs features from the extracted f-NIRs features. DE selects the most significant 50 features from a large dimension (=1100 features) feature sets. 12 best features among 50 optimal features are depicted in Fig. 5 to categorize 4 affective emotions aroused from the hemodynamic response of a subject. It is observed from the figure that the feature f_{93} (mean HbO concentration of channel 4), f_{107} (mean HbO concentration of channel 12), f_{206} (mean HbO concentration of channel 18), f_{273} (mean HbO concentration of channel 20), f_{345} (standard deviation of HbO concentration of channel 15), f_{424} (standard deviation of HbO concentration of channel 16), f_{477} (standard deviation of HbO concentration

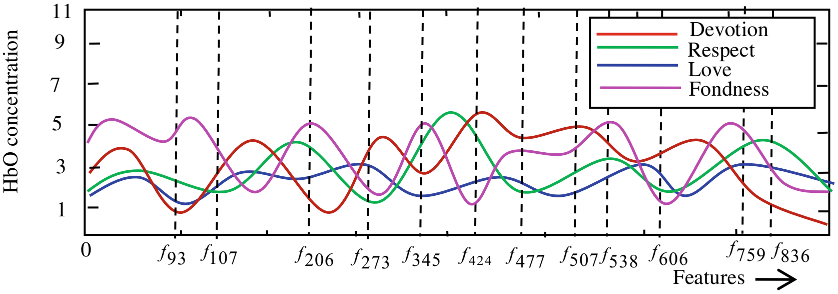
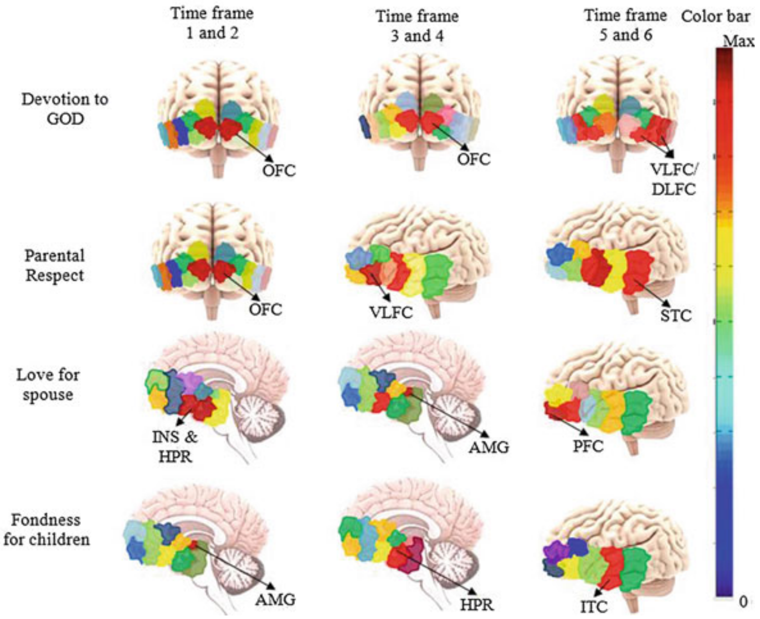


Fig. 5 Feature level discrimination between mean HbO concentrations for four affective emotions

of channel 19), f_{507} (avg. energy of channel 7), f_{538} (avg. energy of channel 10), f_{606} (avg. energy of channel 14), f_{759} (avg. energy of channel 19), f_{836} (skewness of channel 17), have the maximum inter-class separation.

3.3 Experiment 2: Topographic Map Analysis for Individual Emotions

This experiment aims at identifying the corresponding changes in the topographic maps for four individual emotions. Figure 6 illustrates the brain activation regions and their hemodynamic load distribution in brain lobes over different time frames. To capture the temporal features of the cognitive task, the total duration of acquired f-NIRs data has been divided into 6 time frames. It is observed from the plot that the brain activation is shifted from one region to another over different timeframe. For example, the Orbito-frontal cortex (OFC) is highly activated for the first four time-frames then it shifts towards Ventro/Dorso lateral Pre-frontal cortex (VLFC/DLFC) for the emotion aroused due to Devotion. For the emotion of parental respect, the activation shifts from OFC to VLFC and finally, the Superior Temporal cortex (STC) is highly activated in the last two time frames. Similarly, the Insular (INS) and the Hippocampus regions (HPR) are highly activated for the emotion of love and then the activation shifts to the pre-frontal cortex (PFC) through Amygdale (AMG). When the emotion is aroused due to fondness for children, the Amygdale (AMG) region of the temporal lobe is highly activated in the first two time frames and then it shifts to the Hippocampus regions (HPR) and the Inferior Temporal cortex (IFC) for the next time frames.

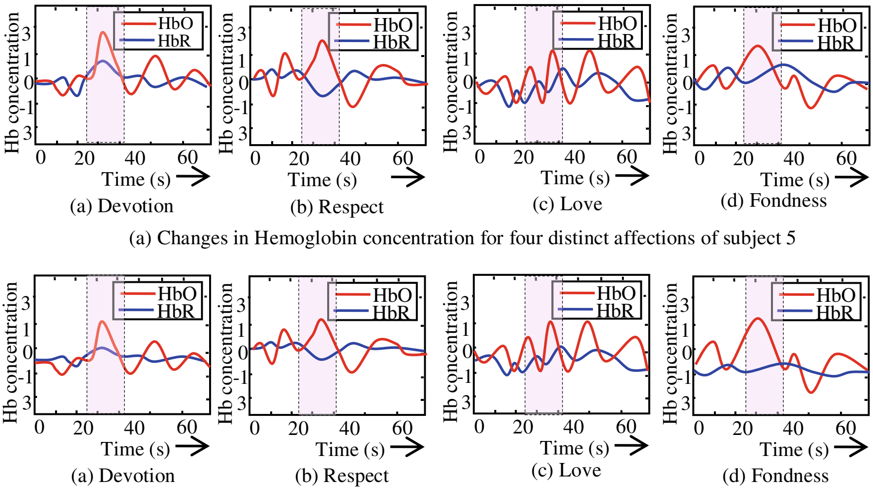


Abbreviations:
OFC = Orbito-frontal cortex, VLFC = Ventro-Lateral Pre-frontal cortex, DLFC = Dorso- Lateral Pre-frontal cortex, STC = Superior Temporal cortex, AMG = Amygdala, HPR = Hippocampus region, ITC = Inferior Temporal Cortex, INS = Insular cortex, PFC = Pre-frontal region.

Fig. 6 Identification of activation regions and their shifts for individual affective emotions in 6 various time frames

3.4 Experiment 3: Variation in Hemoglobin Concentration for Intra and Inter-Subjective Assessment

The prime motivation of this experiment is to determine the intra and inter-subjective variations in oxy-hemoglobin concentration (HbO) and de-oxy-hemoglobin concentration (HbR) over a particular time frame (such as 20 to 40 s) of a selected channel (here, channel 4). It is clearly observed from the experimental results, that the changes in the hemodynamic load take place in the selected time-window for all subjects. Figure 7a, b provide the variation in hemodynamic load distribution for two selected subjects over four distinct emotions. It is apparent from the plot, that the amplitude of oxy-hemoglobin (HbO) and de-oxy hemoglobin (HbR) concentration of subject 5 is increased than subject 9 in the same selected time frame. To minimize this intra-and inter-subjective variations, Type-2 fuzzy classifier is employed in this paper.



(a) Changes in Hemoglobin concentration for four distinct affections of subject 5

(b) Changes in Hemoglobin concentration for four distinct affections of subject 9

Fig. 7 **a** Changes in Hemoglobin concentration for four distinct affections of subject 5. **b** Changes in Hemoglobin concentration for four distinct affections of subject 9

4 Classifier Performance and Statistical Validation

The section deals with the performance analysis of the proposed classifier at four distinct levels. First, the percentage value of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) have been evaluated for each emotion class. Table 1 provides the result of the computation of TP, TN, FP and FN values for a selected (Fondness) class over the existing classifiers. It is apparent from the table that the proposed TSK-based Type-2 Fuzzy classifier yields better performance over its competitors. Second, a comparative study has been undertaken in Table 2 to determine the performance of proposed f-NIRs based classification technique over the EEG- based classification, on the basis of four metrics: Classification Accuracy (CA), Sensitivity (SEN), Specificity (SPE), and F1-score [25].

It is apparent from Table 2 that the performance of the proposed f-NIRs based classifier is enhanced over the EEG-based classification technique by a large margin.

Third, the relative performance of the proposed classifier has been evaluated in Table 3. It is observed from the table that the proposed TSK-based Interval Type-2 Fuzzy classifier outperforms its competitors by a significant level. Finally, the well-known Mc-Nemar's test [32] has been performed for statistical evaluation. According to the Mc-Nemar's test, the value of z-score can be defined as

$$z = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{14}$$

Table 1 Comparative study of the proposed classifier over existing ones

Classifiers	TP%	TN%	FP%	FN%
LSVM classifier [26]	79.0	78.9	21.1	21.0
KSVM-RBF Kernel classifier [27]	82.7	80.8	19.2	17.3
KSVM- polynomial kernel [28]	83.3	84.6	15.4	16.7
BPNN [29]	87.1	88.8	11.2	12.9
Genetic Algorithm based Type-1 Fuzzy classifier [30]	78.1	76.8	23.2	21.9
Difference Evolution (DE) based IT2Fs classifier [31]	90.0	91.8	8.2	10.0
Type-2 fuzzy-RBF- perception neural net (T2F-RBF-PNN) [14]	98.3	97.0	3.0	1.7
Mamdani-based IT2FS [16]	95.5	96.2	3.8	4.5
Proposed TSK-based IT2FS	98.9	97.7	2.3	1.1

Table 2 Comparative performance of EEG [5] and proposed f-NIRs based classification accuracy of affective emotions

Affective emotion classes	EEG based classification accuracy [5]				f-NIRs based classification accuracy (proposed)			
	CA (%)	SPE	SEN	F1-score (%)	CA (%)	SPE	SEN	F1-score (%)
Devotion	78.9	0.78	0.79	78.3	92.6	0.93	0.92	92.5
Respect	79.9	0.85	0.75	79.5	94.7	0.95	0.94	94.8
Love	83.7	0.80	0.85	82.9	95.7	0.96	0.95	96.0
Fondness	82.5	0.85	0.82	82.9	93.5	0.94	0.92	93.7

where, n_{01} denotes the number of classes misclassified by the proposed classification algorithm X but not by the other standard classification algorithm Y. Similarly, n_{10} denotes the number of classes misclassified by Y but not by X. The result of statistical validation is omitted here for space limitation. It is confirmed from the above analysis that the null hypothesis for the standard classifiers are rejected as the z-score of all the other classifiers exceed $\chi^2_{1,0.95} = 3.84$.

5 Conclusion

The paper introduced a novel approach to affective emotion classification using hemodynamic brain response by employing the TSK-based IT-2 Fuzzy classifier. The proposed design requires adaptation of 2 weights w_j and $(1 - w_j)$ for each rule j, which are optimally selected during the training phase to maximize the classification accuracy of all the affection classes. In the test phase, the pre-trained classifier is utilized to classify unknown instances of brain hemodynamic responses corresponding to test data for 4 classes: devotion, respect, love and fondness. Experiments

Table 3 Mean classification accuracy in percentage (standard deviation) of classifiers for 4 distinct emotions

Classifier used	Classification accuracy (Standard Deviation) for four affective emotions			
	Devotion	Respect	Love	Fondness
LSVM classifier[26]	66.7 (0.049)	68.2 (0.042)	66.8 (0.044)	68.0 (0.045)
KSVM-RBF Kernel classifier [27]	71.2 (0.039)	70.9 (0.044)	71.3 (0.034)	70.4 (0.035)
KSVM- polynomial kernel [88]	73.8 (0.039)	74.6 (0.022)	73.3 (0.029)	73.6 (0.055)
BPNN [29]	76.6 (0.049)	77.2 (0.042)	76.8 (0.044)	78.3 (0.045)
Genetic Algorithm based Type-1 Fuzzy classifier [30]	65.2 (0.069)	65.8 (0.062)	65.1 (0.064)	66.4 (0.064)
Difference Evolution (DE) based IT2Fs classifier [31]	80.5 (0.029)	81.3 (0.044)	81.7 (0.056)	81.1 (0.055)
type-2 fuzzy-RBF- perception neural net (T2F-RBF-PNN) [14]	89.2 (0.020)	89.7 (0.022)	90.6 (0.024)	89.5 (0.028)
Mamdani-based IT2FS [16]	91.8 (0.015)	92.7 (0.016)	92.5 (0.011)	92.0 (0.014)
Proposed TSK-based IT2FS	95.2 (0.009)	95.6 (0.008)	95.0 (0.011)	96.3 (0.011)

undertaken confirm the superiority of the said technique over the state-of-the-art techniques, including both classical fuzzy, Type-2 Mamdani based fuzzy and non-fuzzy standard techniques. The proposed TSK-based fuzzy classifier would find interesting applications in sensitivity assessment of healthy and psychiatric disordered people.

References

1. Field T (2010) Touch for socioemotional and physical well-being: a review. Dev Rev 30(4):367–383
2. Bassi G, Gabrielli S, Donisi V, Carbone S, Forti S, Salcuni S (2021) Assessment of psychological distress in adults with type 2 diabetes mellitus through technologies: literature review. J Med Internet Res 23(1):e17740
3. Hatfield E, Rapson RL (2009) The neuropsychology of passionate love. Nova Science Publishers, Psychology of Relationships
4. Cacioppo S, Bianchi-Demicheli F, Hatfield E, Rapson RL (2012) Social neuroscience of love. Clin Neuropsych 9(1)
5. Laha M, Konar A, Rakshit P, Nagar AK (2018)EEG-analysis for classification of touch-induced affection by type-2 fuzzy sets, In: 2018 IEEE symposium series on computational intelligence (SSCI), pp 491–498. IEEE
6. Gullmar D et al (2006) Influence of anisotropic conductivity on EEG source reconstruction: investigations in a rabbit model. IEEE Trans Biomed Eng 53(9):1841–1850

7. Phan KL, Wager T, Taylor SF, Liberzon I (2002) Functional neuroanatomy of emotion: a meta-analysis of emotion activation studies in PET and fMRI. Neuroimage 16(2):331–348
8. Naseer N, Hong KS (2015) fNIRS-based brain-computer interfaces: a review. Front Hum Neurosci 9:3
9. Karnik NN, Mendel JM, Liang Q (1999) Type-2 fuzzy logic systems. IEEE Trans Fuzzy Syst 7(6):643–658
10. Ahmed S, Shakev N, Topalov A, Shiev K, Kaynak O (2012) Sliding mode incremental learning algorithm for interval type-2 Takagi–Sugeno–Kang fuzzy neural networks. Evol Syst 3(3):179–188
11. Zhang Y, Ishibuchi H, Wang S (2017) Deep Takagi–Sugeno–Kang fuzzy classifier with shared linguistic fuzzy rules. IEEE Trans Fuzzy Syst 26(3):1535–1549
12. Tavoosi J, Badamchizadeh MA (2013) A class of type-2 fuzzy neural networks for nonlinear dynamical system identification. Neural Comput Appl 23(3):707–717
13. Wu D, Lin CT, Huang J, Zeng Z (2019) On the functional equivalence of TSK fuzzy systems to neural networks, mixture of experts, CART, and stacking ensemble regression. IEEE Trans Fuzzy Syst 28(10):2570–2580
14. Laha M, Konar A, Rakshit P, Ghosh L, Chaki S, Ralescu AL, Nagar AK (2018)Hemodynamic response analysis for mind-driven type-writing using a type 2 fuzzy classifier. In: 2018 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp 1–8. IEEE, July
15. Ghosh L, Konar A, Rakshit P, Nagar AK (2018) Hemodynamic analysis for cognitive load assessment and classification in motor learning tasks using type-2 fuzzy sets. IEEE Trans Emerg Topics Comput Intell 3(3): 245–260
16. Laha M, Konar A, Rakshit P, Nagar AK (2019) Exploration of subjective color perceptual-ability by eeg-induced type-2 fuzzy classifiers. IEEE Trans Cogn Dev Syst 12(3):618–635
17. Kachenoura A, Albera L, Senhadji L, Comon P (2008) ICA: a potential tool for BCI systems. IEEE Signal Process Mag 25(1):57–68
18. Chowdhury E, Qadir Z, Laha M, Konar A, Nagar AK (2020)Finger-induced motor imagery classification from hemodynamic response using type-2 fuzzy sets. In: Soft computing for problem solving 2019, pp 185–197. Springer, Singapore
19. Das S, Abraham A, Konar A (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. Advances of computational intelligence in industrial systems, Springer, Berlin, Heidelberg, pp 1–38
20. Rakshit P, Konar A, Das S (2017) Noisy evolutionary optimization algorithms–a comprehensive survey. Swarm Evol Comput 33:18–45
21. Kreyszic E (2000) Advanced engineering mathemetics. Wiley
22. Jang JSR, Sun CT, Mizutani E (1997) Neuro-fuzzy and soft computimg. Prentice-Hall
23. Laha M, Konar A, Das M, Debnath C, Sengupta N, Nagar AK (2020)P200 and N400 induced aesthetic quality assessment of an actor using type-2 fuzzy reasoning, In: 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp 1–8. IEEE, July
24. De A, Laha M, Konar A, Nagar AK (2020)Classification of relative object size from parietooccipital hemodynamics using type-2 fuzzy sets. In: 2020 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp. 1–8. IEEE, July
25. Al-Salman W, Li Y, Wen P (2019) Detection of EEG K-complexes using fractal dimension of time frequency images technique coupled with undirected graph features. Front Neuroinform 13:45
26. Wang W, Xu Z, Lu W, Zhang X (2003) Determination of the spread parameter in the Gaussian kernel for classification and regression. Neurocomputing 55(3,4):643–663
27. Goodale CL, Aber JD, Ollinger SV (1998) Mapping monthly precipitation, temperature, and solar radiation for Ireland with polynomial regression and a digital elevation model. Climate Res 1:35–49
28. Dao VN, Vemuri VR (2002)A performance comparison of different back propagation neural networks methods in computer network intrusion detection. Differ Equ Dyn Syst 10(1&2):201–214

29. Bhattacharya D, Konar AA, Das P (2016) Secondary factor induced stock index time-series prediction using self-adaptive interval type-2 fuzzy sets. Neurocomputing 171:551–568
30. Basu D, Bhattacharyya S, Sardar D, Konar A, Tibarewala DN, Nagar AK (2014) A differential evolution based adaptive neural Type-2 Fuzzy inference system for classification of motor imagery EEG signals. In: FUZZ-IEEE, pp 1253–1260
31. Ghosh L, Konar A, Rakshit P, Nagar AK (2019) Mimicking short-term memory in shape-reconstruction task using an EEG-induced type-2 fuzzy deep brain learning network. IEEE Trans Emerg Topics Comput Intell 4(4):571–588
32. Sun X, Yang Z (2006) Generalized McNemars Test for homogeneity of the marginal distribution. In: Proceedings of SAS global Forum, paper 382

Brain–Computer Interface for Fuzzy Position Control of a Robot Arm by Mentally Detected Magnitude and Sign of Positional Error

Arnab Rakshit and Amit Konar

Abstract The paper addresses a novel approach to position control of a robot arm by utilizing three important brain signals, acquired with the help of an EEG interface. First, motor imagery signal is employed to activate the motion of a robotic link. Second, the error-related potential signal is acquired from the brain to stop the motion of the robotic link, when it crosses a predefined target position. Third, the approximate magnitude of the positional error is determined by steady-state visual evoked potential signal, acquired by noting the nearest flickering lamp that the robotic link has just crossed. The novelty of the present research is to decode the approximate magnitude of the positional error. Once the approximate magnitude and sign of the positional errors are obtained from the mental assessment of the experimental subject, the above two parameters are fed to a fuzzy position controller to generate necessary control commands to control the position of the end-effector of the robotic link around the predefined target position. Experiments undertaken confirm a low percentage of overshoot and small settling time of the proposed controller in comparison to those published in the current literature.

Keywords EEG · Robotic arm · ERD/ERS · ErrP · SSVEP · Fuzzy control

1 Introduction

Brain–computer interface (BCI) is currently gaining increasing potential for its widespread applications in rehabilitative robotics. People with neuro-motor disability such as Amyotrophic Lateral Sclerosis (ALS), partial paralysis, and the like require assistive support to perform their regular day-to-day works, such as delivery of food

A. Rakshit (✉) · A. Konar
Department of Electronics & Tele-communication Engineering, Jadavpur University, Kolkata, India
e-mail: arnabrakshit2008@gmail.com

A. Konar
e-mail: konaramit@yahoo.co.in

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_13

[1], medicines [2], etc. by an artificial robotic device, where the patients themselves can control the movements of the robot arm, their pick-up, placements, etc. by mind-generated control commands. Neuro-prosthesis is one of the most active areas of BCI research for its inherent advantage to rehabilitate people with degenerative neuro-motor diseases. Early research on neuro-prosthetics began with the pioneering contribution of Pfurtscheller [3, 4], who experimentally could first demonstrate the scope of one fundamental brain signal, called Motor Imagery, technically titled as Event-Related Desynchronization followed by Event-Related Synchronization (ERD/ERS). This signal appears in the motor cortex region of a person, when he/she thinks of moving his/her arms/legs or any voluntarily movable organs. Several researchers have utilized this signal for mind-driven motion-setting to a mobile robot [5], local navigating device [6, 7], artificial robotic arm [8–10], and many others. However, using ERD/ERS signal alone can switch on or switch off a device, and thus can only be used for open-loop applications.

In order to utilize the ERD/ERS in closed-loop position control applications, we need additional brain signals. Several research groups [11–13] have taken initiatives to utilize the benefits of Error-Related Potential (ErrP) and/or P300 signals to develop a generic platform for closed-loop position control applications. It is important to mention here that the ErrP signal is liberated from the z-electrodes, located at the midline of our scalp, when a subject himself commits any motion-related error and/or finds a second person or a machine to commit similar errors. The ERD/ERS and ErrP signals have been employed in a number of robot position control systems to set in motion of the robotic motor on emergence of the MI (ERD/ERS signal) and switch off the motor of the robot arm, when the robotic link crosses a fixed target position. However, the primary limitation of such position control schemes is on–off control strategy, which according to classical control theory results in large steady-state error [14].

To overcome this problem, several extensions to the basic control strategies have been proposed in the recent past [13, 15]. In [15], the authors developed a new strategy to reduce large steady-state error by commanding the robot to turn in reverse direction at a relatively lower speed than its current speed and also sensing the second, third P300, when the target is crossed several times by the end-effector. Such scheme can result in reduced steady-state error but at the cost of extra settling time.

The present research can reduce both steady-state error and settling time as it happens to be in case of classical control strategy by assessing the sign and magnitude of positional error from the subject's brain. However, as the magnitude of error is approximate, a fuzzy controller is a more appropriate option in contrast to a traditional controller. A set of fuzzy rules are proposed to infer the position of the end-effector from the approximate magnitude and sign of positional errors. Traditional Mamdani-type fuzzy reasoning is employed to yield the fuzzified end-effector positions. In case a number of fuzzy rules fire synchronously, the union of the inferences is considered. Finally, a defuzzifier is used to get back the controlled position of the end-effector. The proposed approach thus is unique and remained unknown to the BCI research community.

The paper is divided into five sections. In Sect. 2, we provide the principles adopted for position control using magnitude and sign of error, captured from the acquired ErrP and SSVEP signals. Section 3 deals with a discussion on processing of the acquired brain signals to make them free from noise and extraction of certain features from the pre-processed signals for classification. Section 4 deals with fuzzy controller design. Section 5 covers the experimental issues and also narrates the main results justifying the claims. A list of conclusions is included in Sect. 6.

2 Principles Adopted in the Proposed Position Control Scheme

This section provides the principles of position control using three brain signals: (i) motor imagery to actuate the motion of a robotic link, (ii) stopping the robotic link by sensing the ErrP signal, and (iii) assessing the magnitude of positional error from the flickering Light Emitting Diode (LED) closest to the stopping position. It is indeed important to mention here that assessment of the magnitude of error by SSVEP introduced here is novel and primary contribution of the present research. The sign and the magnitude of positional error together help in generating the accurate control action for the position control application. The principles of the BCI-based position control scheme are given in Fig. 1. It is noteworthy from Fig. 1 that the controller receives both sign and magnitude of error to generate the control signal. However, the exact measure of the magnitude of error cannot be performed easily for practical limitation in placement of SSVEP sources continuously along the trajectory of motion of the robotic end-effector. To overcome the present problem, an approximate assessment of the positional error is evaluated in five scales: NEAR ZERO, SMALL POSITIVE, LARGE POSITIVE, SMALL NEGATIVE and LARGE NEGATIVE using fuzzy membership functions [16]. The control signal about position of the end-effector is also fuzzified in the same five scales. Such assessment helps in generating fuzzy inferences about the degree of memberships of control signals in multiple fuzzy sets. It is indeed important to mention here that a fuzzy system usually is much robust in comparison to traditional rule-based expert systems as it takes care of aggregation of the inferences obtained from firing of multiple rules simultaneously by taking fuzzy union of the generated inferences. The defuzzification of the overall inference returns the signal back in the real domain. There exist several defuzzification procedures. Here, the center of gravity (CoG) defuzzification is used for its simplicity and wide popularity in fuzzy research community [17].

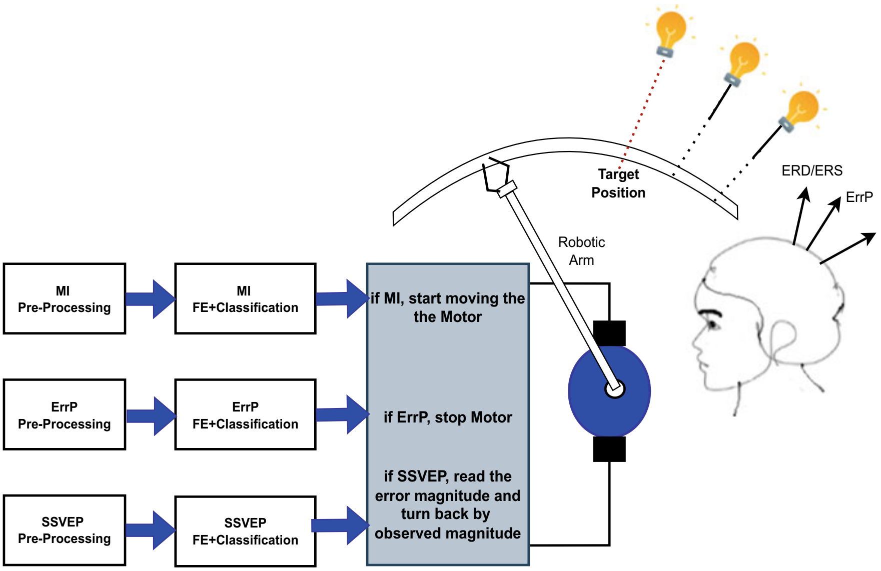


Fig. 1 Overview of BCI-based position control scheme

3 Signal Processing and Classification of Brain Signals

This section provides an overview of the basic signal processing, feature extraction, and classifier design aspects for the proposed application.

3.1 ERD-ERS Feature Extraction and Classification

For ERD-ERS feature extraction, we need to take as many as 500 offline instances of motor imagery (MI) signals acquired from the motor cortex regions of the subject. These 500 instances of MI signals are examined manually to identify around 300 true positive (v-shaped) and around 200 false negative (non-V or V with inadequate depth) instances. Both the true positive and false negative instances are then sampled at a fixed interval of time, and the mean and variance of the signals at each sampled point are evaluated. Let, at a given sample point s_i, we obtain 300 values from 300 true positive curves. Now a Gaussian model is constructed for each sample point s_i, with mean $= m_i$ and standard deviation σ_i. The sample values that lie within $m_i \pm 3\sigma_i$ are used and the rest are discarded. Thus, for each time position in the training samples, we accommodate selected values of the existing trials. Similarly, we undertake selective sample values from a pool of 200 EEG false negative instances. These true positive and false negative instances of the ERD/ERS signals are used subsequently to train a classifier. In this paper, Common Spatial Pattern (CSP), which is widely used in BCI literature as an optimized spatial filter [18], is employed to evaluate the data covariance matrices for the two classes to effectively project the training samples into CSP features. These CSP features are then transferred to a two-level classifier to recognize the positive and negative motor imagery (MI) signals.

For classification of MI and resting conditions (also called NO motor imageries), the following steps are followed. Let X_1 and X_2 be $m \times n$ matrices, where m and n, respectively, denote number of EEG channels and number of time samples. Let $C1$ and $C2$ be the spatial covariance matrices given by $C1 = X_1 X_1^T$and $C2 = X_2 X_2^T$ for positive (MI) and false negative classes. The motivation of CSP is to obtain filter vector **w**, such that the scalar $\mathbf{wC_1w^T/wC_2w^T}$ is maximized. Once optimal value of vector w is evaluated, the variances of CSP projections $\mathbf{wX_1}$ and $\mathbf{wX_2}$ are utilized as CSP features of two classes. Any traditional linear classifier, such as Linear Discriminant Analysis (LDA) or Linear Support Vector Machine (LSVM), and the like can be used for classification of the MI signals from the resting states. Here, the authors employed Kernelized Support Vector Machine (KSVM) with Radial Basis Function (RBF) kernel for its proven accuracy in high-dimensional non-linear classification [19].

3.2 *ErrP Feature Extraction and Classification*

Previous research on ErrP feature extraction reveals that the characteristics of ErrP signal can be better captured by time-domain parameters, such as Adaptive Autoregressive (AAR) coefficients [13]. This inspired the authors to utilize AAR features for the detection of ErrP. In the present research, AAR parameters are extracted from approximately 500 ErrP instances and 500 resting states in offline training phase. A q-order AAR expresses each EEG sample as a linear combination of past q samples along with an error term characterized by zero mean Gaussian process. AAR coefficients are estimated using Least Mean Square (LMS) algorithm with an update parameter of 0.0006. For an EEG signal of 1s duration (200 samples), a 6th-order AAR generates $6 \times 200 = 1200$ AAR parameters which are used as the feature vector of the EEG trial. An LSVM classifier is then developed to determine the unique set of weights of the classifier to classify the ErrP and non-ErrP instances in real time.

3.3 *SSVEP Detection*

For detection of SSVEP, the occurrence of the peak power at the flickering frequency of the stimulus is checked. To test this, the maximum power in the PSD is searched over the frequency spectrum. If there is a single peak power occurring at the flickering frequency, then SSVEP is confirmed. In this study, we estimated the spectral power density through Welch's modified periodogram method [20]. Power spectral density is obtained for each stimulus frequency and their first two harmonics. We considered an interval 1 Hz below and above the stimulus frequency to obtain the PSD. Once the PSD values associated to each SSVEP stimulus frequency are obtained, we search for the frequency that has highest PSD value. The frequency having the highest frequency value is considered as the target stimulus.

4 Fuzzy Controller Design

The novelty of the current paper is to determine the controller response from the approximate measure of magnitude of error. Here, the occurrence of the error signal is determined from the occurrence of ErrP signal. Now, to measure the magnitude of the error signal, a set of flickering light sources are placed at regular intervals. All these sources flicker at disjoint frequencies. When the subject observes the robotic arm crossing the target position, he is supposed to yield an ErrP signal from the z-electrodes. Almost simultaneously, he is supposed to release an SSVEP signal. Generally, people suffering from neuro-motor diseases have relatively poor reflex, and so they take longer time to respond to flickering visual signals. In order to alleviate this problem, light sources flickering at different frequencies are placed around their trajectory of the end-effector. Here, the subject has to pay attention to the nearest flickering source, close enough to the terminal position of the end-effector. Here, the flickering signal of the sources has frequencies in the ascending order of their distances from the predefined target position. This makes sense in the way that larger is the distance of the flickering source from the target position, the larger is the frequency of the source. A set of fuzzy quantifiers is employed to quantify the measure of the positional error in five grades: NEAR ZERO(NZ), SMALL POSITIVE(SP), LARGE POSITIVE(LP), SMALL NEGATIVE(SN) and LARGE NEGATIVE(LN). A knowledge base comprising a set of rules that map the fuzzified errors into fuzzy control signals is then utilized to derive the control signals for each fired rule. The union of the fuzzy control signals is taken, and the result is defuzzified to get back the actual value of the control signal.

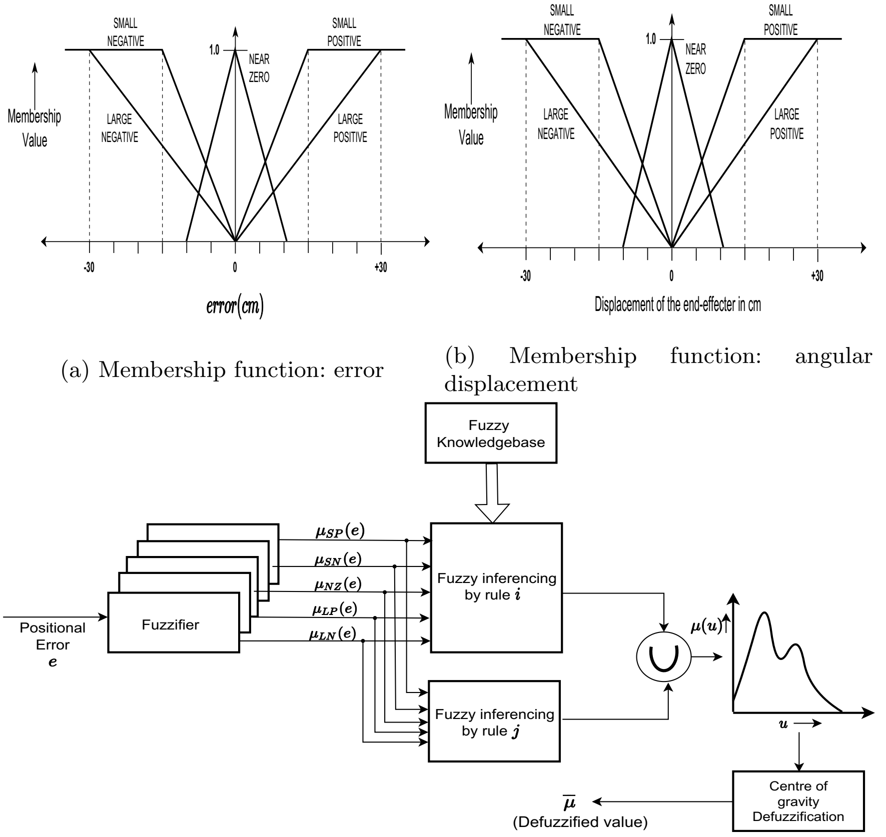
4.1 Fuzzy Reasoning in the Control Problem

Consider the fuzzy production rules:

Rule 1: If x is A_1 then y is B_1
Rule 2: If x is A_2 then y is B_2

$\vdots$

Rule n: If x is A_n then y is B_n



(a) Membership function: error

(b) Membership function: angular displacement

(c) Architecture of the proposed fuzzy controller

Fig. 2 Architecture of the proposed fuzzy controller and schematic overview of membership curves

Here x,y are linguistic variables in the universes X and Y, respectively. A_1, A_2, $\ldots$, A_n are fuzzy sets under the universe X and B_1, B_2, $\ldots$, B_n are fuzzy sets under the universe Y. Let $x = x'$ be a measurement. We compute the fuzzy inference for the given measurement $x = x'$ by the following steps:
Step 1: Compute: $\alpha_1 = Min(\mu_{A_1}(x'), \mu_{B_1}(y))$, $\alpha_2 = Min(\mu_{A_2}(x'), \mu_{B_2}(y))$,..., $\alpha_n = Min(\mu_{A_n}(x'), \mu_{B_n}(y))$.
Step 2: Evaluate the overall fuzzy inference $\mu_{B'}(y) = Max(\alpha_1, \alpha_2, \ldots, \alpha_n)$. After the fuzzy inference $\mu_{B'}(y)$ is evaluated, we compute the centroid of it by "center of gravity" method [21].

In the present control problem, x is error and y is displacement of the end-effector. The fuzzy rules constructed for the position control system are triggered appropriately depending on magnitude and sign of error signal and the selected rules on firing generate inferences, the union of which is the resulting control signal, representing displacement of the end-effector. The fuzzy membership functions involving error are SMALL POSITIVE, etc. and angular displacement are SMALL NEGATIVE, etc. which are given in Fig. 2a, b and architecture of the proposed fuzzy controller is given in Fig. 2c. The list of fuzzy rules used for the generation of control signals is given below:

Rule 1: If error is SMALL POSITIVE then angular displacement is SMALL NEGATIVE.
Rule 2: If error is SMALL NEGATIVE then angular displacement is SMALL POSITIVE.
Rule 3: If error is NEAR ZERO then angular displacement is NEAR ZERO.
Rule 4: If error is LARGE NEGATIVE then angular displacement is LARGE POSITIVE.
Rule 5: If error is LARGE POSITIVE then angular displacement is LARGE NEGATIVE.

5 Experiments and Results

This section first describes the experimental protocol in a detailed way and represents the major outcomes of the experiment in subsequent stages. Key details of the experiment are highlighted below.

5.1 Subjects

Twelve people within a age group of 18–40 years (mean age 32) voluntarily participated in the study. None of them had any prior experience with BCI training. Out of the twelve volunteers, 6 were male, 6 were female, and 2 of them were differently abled (Sub11 and Sub12). The objective and procedure of experiment were made clear to the volunteers before conducting the experiment and a consent form stating their willingness to participate in the study was duly signed by them. The experiment was conducted in adherence to the Helsinki Declaration 1970 later revised in 2000 [22].

5.2 EEG System

EEG data were acquired from the volunteers using a 19 channel EEG amplifier device made by the company Nihon-Kohden. The EEG system has sampling rate 200 Hz and comes with built-in notch filter 50 Hz frequency. EEG electrodes were placed over the scalp by following the international 10–20 electrode placement convention

[23]. Out of the total 19 electrodes, we used six electrode positions (C_3, C_4, C_z over the motor cortex and P_3, P_4, Pz over the parietal lobe) to acquire the Motor Imagery brain signals. For the SSVEP and ErrP brain signals, we used $\{O_1, O2\}$ and $\{Fz, P_z\}$ electrode positions, respectively.

5.3 Training Session

We conducted the training session throughout the 15 d with a repetitions of 3 sessions in a day for each subject. Inter-session gap of 10 min was provided. Each session consists of 50 trials, resulting 150 trials for a subject in a day. Each trial contains the visual instruction to be followed by the participating subjects.

Visual instructions are presented before the subject through a robotic simulator. The robotic simulator virtually represents a robotic limb capable of producing clock/anti-clockwise movement around a specially designed fixed frame. The frame has markings of various positions over it along with the target position and LEDs are mounted near the frame against each positional markings. The LEDs flicker with a constant frequency but are different from each other.

A trial starts with a fixation cross that appears as a visual cue and asks the subject to remain alert for the upcoming visual cues. It stays on the screen for 2s duration. The next visual cue contains an instruction to perform either LEFT or RIGHT arm motor imagery for clockwise/anti-clockwise movement of the robotic limb. The next visual cue contains a scenario where the moving link commits an error by crossing the target location, hence the subject develops ErrP brain pattern by observing the error. The next scenario illustrates a condition where the end-effector of the moving link crossed the target position. Now, Subjects are instructed to focus their gaze on the flickering LED nearest to the present position of robot end-effector, focusing on the flickering source which generates an SSVEP signal modulated by the source frequency in the subjects' brain. Timing diagram of stimulus presentation is depicted in Fig. 3.

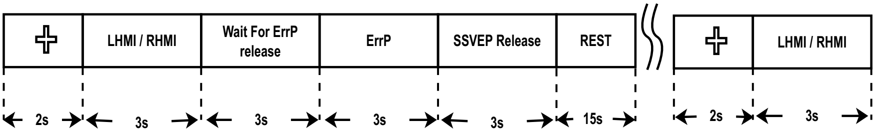


Fig. 3 Stimuli diagram of training session

5.4 Testing Session

The major difference between training session and testing session lies in the medium of operation. In contrast to the training session, which is conducted offline using a robotic simulator, the testing session is performed in real time with the physical robot. This session is more complex than training session as the subject participating in this session does not receive any visual instruction to perform the required mental task. Hence, the subjects need to plan the three steps of action (viz., link movement, target selection, and gazing on the nearest flickering source) themselves without any visual guidance.

A timing diagram presented in Fig. 4 shows the time taken by each module during real-time operation. During the real-time operation, we used a window of 1s duration to acquire the MI signal and SSVEP signal, whereas ErrP was acquired through the windows of 250 ms.

5.5 Results and Discussions

The results of the current experiment are presented in three stages. First, we provide a comparative analysis between the performance of the proposed feature extraction and classifier combination and other widely used methods in BCI literature. The performance is evaluated by averaging the performance of all the subjects over all the sessions during the testing phase. In the second stage, we provide performance analysis of all the subjects that participated in the testing session, and the performance of the proposed fuzzy controller is presented in the third stage.

Performance of the brain signal detection methods is evaluated on the basis of four metrics—Classification Accuracy (CA), True Positive Rate (TPR), False Positive Rate (FPR), and Cohen's kappa index (κ) as used in [15].

Performance of MI detection is presented in the first phase of Table 1. Along with the proposed Feature Extraction and Classifier combination (CSP + RBF SVM), we considered six other combinations to compare the performance. It is evident from the table that the proposed feature extraction+classifier combination worked best in our case yielding an average accuracy of 91.31% with average TPR, FPR, and kappa of 0.89, 0.04, and 0.84, respectively.

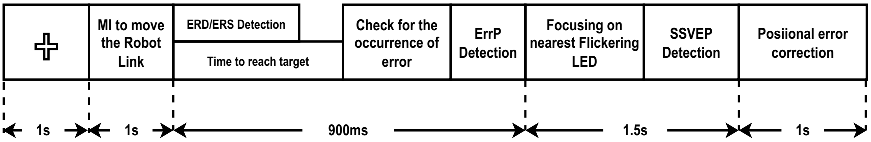

MI to move the Robot Link
ERD/ERS Detection
Time to reach target
Check for the occurrence of error
ErrP Detection
Focusing on nearest Flickering LED
SSVEP Detection
Posiional error correction
1s
1s
900ms
1.5s
1s

Fig. 4 Timing diagram of testing session

Table 1 Comparative study of different ERP detection methods

Brain pattern detection	Feature extraction + classifier	Performance metrics			
		CA (%)	TPR	FPR	kappa
MI detection	CSP+KSVM-RBF	91.31	0.89	0.04	0.84
	CSP+LSVM	90.11	0.89	0.05	0.83
	CSP+QDA	87.19	0.85	0.06	0.79
	DWT+KSVM-RBF	84.45	0.83	0.07	0.84
	DWT+QDA	88.56	0.89	0.05	0.80
	Hjorth+KSVM-RBF	82.38	0.81	0.09	0.75
	Hjorth+QDA	80.62	0.80	0.09	0.72
ErrP classifier	AAR+LSVM	92.71	0.91	0.04	0.82
	AAR+LDA	90.18	0.85	0.06	0.80
	Temporal Feature+ANN	83.13	0.82	0.07	0.76
	Temporal Feature+LDA	80.52	0.79	0.08	0.74
	SWLDA	91.23	0.90	0.04	0.81
SSVEP classifier	PSD(Welch)+Threshold	92.89	0.92	0.04	0.86
	PSD(Welch)+LSVM	93.80	0.93	0.03	0.85
	FFT	88.81	0.87	0.05	0.78
	CCA	94.96	0.94	0.02	0.88

CSP = Common Spatial Pattern
KSVM-RBF = Kernelized Support Vector Machine with Radial basis function kernel
LSVM = Linear Support Vector machine, DWT = Discrete Wavelet Transform
QDA = Quadratic Discriminant Analysis, LDA = Linear Discriminant analysis
ANN = Artificial Neural Network, CCA = Canonical Correlation Analysis

ErrP detection and SSVEP detection performances are compared with other relevant methods and results are presented in the second and third phases of Table 1. It is observed that average ErrP detection accuracy is achieved as high as 92% followed by the TPR, FPR, and kappa of 0.91, 0.04, and 0.82. Clearly, the present ErrP detection scheme outperforms the other methods by a significant margin.

We see a similar result in SSVEP performance, where the present SSVEP detection method achieves a moderately high detection accuracy of 93% with the TPR=0.92, FPR=0.04, and kappa=0.86. Although CCA here performs a little better than our proposed detection method, still we choose the proposed method for the major advantage of being computationally very inexpensive, hence most suitable for real-time operation.

Performances of all the subjects participated in the experiment are given in Tables 2, ,3, and 4. Each participant is evaluated through four metrics (CA, TPR, FPR, and kappa(κ)) described earlier. Average classification time taken by the clas-

Table 2 Subjectwise motor imagery detection result

Subject	Performance metrics (MI Detection)				
	CA% ± std	TPR	FPR	Kappa(κ)	Time(s)
Sub1	92.82±2.39	0.92	0.03	0.86	0.602
Sub2	93.96±1.82	0.92	0.03	0.91	0.549
Sub3	94.39±1.06	0.93	0.02	0.92	0.553
Sub4	89.81±2.21	0.86	0.03	0.81	0.608
Sub5	87.84±1.89	0.86	0.06	0.84	0.574
Sub6	94.49±1.84	0.92	0.04	0.89	0.579
Sub7	92.16±1.95	0.91	0.03	0.81	0.601
Sub8	91.87±1.26	0.89	0.04	0.83	0.583
Sub9	94.23±1.93	0.93	0.03	0.81	0.559
Sub10	93.12±1.28	0.93	0.05	0.84	0.552
Sub11	86.82±4.28	0.87	0.08	0.78	0.548
Sub12	84.23±3.73	0.85	0.07	0.76	0.571

Table 3 Subjectwise ErrP detection result

Subject	Performance metrics (ErrP detection)				
	CA% $\pm std$	TPR	FPR	Kappa(κ)	Time(s)
Sub1	94.81± 1.05	0.93	0.03	0.82	0.109
Sub2	94.52±1.01	0.94	0.04	0.90	0.113
Sub3	91.86±2.09	0.90	0.03	0.79	0.108
Sub4	93.47±0.98	0.92	0.04	0.81	0.121
Sub5	94.31±0.77	0.95	0.04	0.78	0.111
Sub6	93.28±1.46	0.92	0.03	0.93	0.107
Sub7	90.63±2.58	0.91	0.03	0.81	0.118
Sub8	89.86±2.81	0.88	0.03	0.85	0.105
Sub9	92.19±1.63	0.90	0.02	0.86	0.118
Sub10	90.25±2.28	0.90	0.06	0.78	0.108
Sub11	89.11±3.13	0.90	0.06	0.79	0.113
Sub12	86.28±2.08	0.85	0.05	0.72	0.110

sifier during the testing time is also reported in the above tables. Table 2 reveals that the highest detection accuracy of MI brain pattern is achieved for the sixth subject (CA=94.49%) while the third subject shows the highest kappa value of 0.92 indicating highest reliability. As revealed from Tables 3 and 4, the other two brain patterns, ErrP and SSVEP, were detected with maximum accuracy of 94.81% and 95.27%, respectively. The highest ErrP accuracy is observed with the first subject while the fifth subject shows the highest SSVEP accuracy. For the above two categories of signal, the highest kappa values are achieved as 0.93 and 0.92.

Table 4 Subjectwise SSVEP detection result

Subject	Performance metrics(SSVEP detection)				
	CA% ± std	TPR	FPR	Kappa(κ)	Time(s)
Sub1	93.88±0.89	0.92	0.02	0.91	0.091
Sub2	91.49±0.96	0.92	0.03	0.88	0.082
Sub3	91.90±0.93	0.90	0.04	0.82	0.095
Sub4	95.06±0.18	0.95	0.03	0.91	0.090
Sub5	95.27±0.27	0.94	0.02	0.92	0.086
Sub6	89.26±2.65	0.90	0.03	0.86	0.097
Sub7	92.43±1.03	0.91	0.03	0.87	0.092
Sub8	90.79±1.88	0.89	0.05	0.82	0.089
Sub9	93.72±0.98	0.94	0.05	0.81	0.103
Sub10	90.93±2.15	0.90	0.05	0.81	0.098
Sub11	85.89±5.05	0.84	0.08	0.72	0.089
Sub12	88.21±4.29	0.89	0.05	0.80	0.085

5.6 Comparison of System Performance

The overall position control performance of the system is evaluated using few popular metrics taken from control system literature. The metrics are success rate, steady-state error (SS error), peak overshoot, and settling time [14, 15].

Overall performance of the system is presented in Table 5. Results are averaged over all the subjects over all the testing sessions. Performance result is compared with five other relevant strategies. First the result is compared with the open-loop control strategy solely based on Motor Imagery [24]. Success rate obtained in this case found to be (76.2%). Next, the proposed method is compared with four hybrid BCI control strategies, where researchers, instead of relying on a single brain pattern, used multiple brain signals to design a robust interface for mentally controlling a robot arm. We considered four different control strategies that used four different combinations of brain signals (MI+SSVEP [25], MI+P300 [26], MI+ErrP [13], and MI+SSVEP+P300 [15]). Comparison results are obtained by implementing the control strategies in our own BCI setup.

It is evident from Table 5 that our proposed method achieves highest success rate (92.1%) among all the control strategies. It also ensured the lowest settling time (6s), steady-state error (0.15%), and peak overshoot (4.1%) among strategies under comparison. Although the present scheme shows improvement over all the fields considered in Table 5, the major improvement is considered to be the drastic reduction of settling time with simultaneous reduction of steady-state error and peak overshoot. Hence, the proposed fuzzy BCI controller outperforms the rest of the control strategies by a significant margin.

Table 5 Relative performance analysis

Strategies	Performance metrics			
	Success	SS	Peak	Settling
	Rate	Error(%)	Overshoot(%)	Time(s)
MI [24]	76.2	6.22	6.2	18
MI+SSVEP [25]	88.5	6.09	5.9	15
MI+P300 [26]	84.3	3.21	4.5	13
MI+ErrP [13]	85.8	2.1	4.9	16
MI+P300+SSVEP [15]	90.2	0.31	4.2	20
Proposed method	92.1	0.15	4.1	6

6 Conclusion

This paper claims to have utilized mentally generated sign and magnitude of positional error for automatic control of artificial robotic limb. The principles and realization of the above idea being novel in the realm of BCI are expected to open up new direction of control strategies, parallel to traditional controllers, as both the (approximate) magnitude and sign of positional error are known beforehand. Because of approximate estimation of positional errors, the logic of fuzzy sets has been incorporated that could handle the approximations and yields good control accuracy with small peak overshoot below 4.1% and settling time around 6 s.

References

1. Ha J, Kim L (2021) A brain-computer interface-based meal-assist robot control system. In: 2021 9th international winter conference on brain- computer interface (BCI). IEEE. 2021, pp 1–3
2. Ha J et al (2021) A hybrid brain-computer interface for real-life meal-assist robot control. Sensors 21(13):4578
3. Pfurtscheller G et al (2003) Graz-BCI: state of the art and clinical applications. IEEE Trans Neural Syst Rehabil Eng 11(2):1–4
4. Pfurtscheller G et al (2000) Current trends in Graz brain-computer interface (BCI) research. IEEE Trans Rehabil Eng 8(2):216–219
5. Liu Y et al (2018) Brain-robot interface-based navigation control of a mobile robot in corridor environments. IEEE Trans Syst Man Cybern: Syst 50(8):3047–3058
6. Tonin L, Bauer FC, Millán JDR (2019) The role of the control framework for continuous teleoperation of a brain-machine interface-driven mobile robot. IEEE Trans Robot 36(1):78–91
7. Chen X et al (2022) Clinical validation of BCI-controlled wheelchairs in subjects with severe spinal cord injury. IEEE Trans Neural Syst Rehabilitation Eng 30:579–589
8. Chen X et al (2019) Combination of high-frequency SSVEP-based BCI and computer vision for controlling a robotic arm. J Neural Eng 16(2):026012

9. Casey A et al (2021) BCI controlled robotic arm as assistance to the rehabilitation of neurologically disabled patients. Disabil Rehabil: Assist Technol 16(5):525–537
10. Vilela M, Hochberg LR (2020) Applications of brain-computer interfaces to the control of robotic and prosthetic arms. Handb Clin Neurol 168:87–99
11. Wang X et al (2022) Implicit robot control using error-related potential-based brain-computer interface. IEEE Trans Cogn Dev Syst
12. Iretiayo A et al (2020) Accelerated robot learning via human brain signals. In: IEEE international conference on robotics and automation (ICRA). IEEE, pp 3799–3805
13. Bhattacharyya S, Konar A, Tibarewala DN (2017) Motor imagery and error related potential induced position control of a robotic arm. IEEE/CAA J Autom Sin 4(4):639–650
14. Nagrath IJ, Gopal M (2007) Control systems engineering. In: New age international publishers, pp 193–268. ISBN: 81-224-2008-7
15. Rakshit A, Konar A, Nagar AK (2020) A hybrid brain-computer interface for closed-loop position control of a robot arm. In: IEEE/CAA J Autom Sin 7(5):1344–1360
16. Starczewski JT (2012) Advanced concepts in fuzzy logic and systems with membership uncertainty, Vol. 284. Springer, Berlin
17. Zimmermann H-J (2011) Fuzzy set theory-and its applications. Springer Science & Business Media
18. Lotte F, Guan C (2010) Spatially regularized common spatial patterns for EEG classification. In: 2010 20th international conference on pattern recognition. IEEE. 2010, pp 3712–3715
19. Bousseta R et al (2016) EEG efficient classification of imagined hand movement using RBF kernel SVM. In: 2016 11th international conference on intelligent systems: theories and applications (SITA). IEEE, pp 1–6
20. Carvalho SN et al (2015) Comparative analysis of strategies for feature extraction and classification in SSVEP BCIs. Biomed Signal Process Control 21 :34–42
21. Konar A (2006) Computational intelligence: principles, techniques and applications. Springer Science & Business Media
22. General Assembly of the World Medical Association et al (2014) World medical association declaration of Helsinki: ethical principles for medical research involving human subjects. J Am Coll Dent 81(3):14–18
23. Homan RW, Herman J, Purdy P (1987) Cerebral location of international 10–20 system electrode placement. Electroencephalogr Clin Neurophysiol 66(4):376–382
24. Bousseta R et al (2018) EEG based brain computer interface for controlling a robot arm movement through thought. Irbm 39(2):129–135
25. Yan N et al (2019) Quadcopter control system using a hybrid BCI based on off- line optimization and enhanced human-machine interaction. IEEE Access 8:1160–1172
26. Yu Y et al (2017) Self-paced operation of a wheelchair based on a hybrid brain-computer interface combining motor imagery and P300 potential. IEEE Trans Neural Syst Rehabil Eng 25(12):2516–2526

Social Media Sentiment Analysis on Third Booster Dosage for COVID-19 Vaccination: A Holistic Machine Learning Approach

Papri Ghosh, Ritam Dutta, Nikita Agarwal, Siddhartha Chatterjee, and Solanki Mitra

Abstract Over a period of more than two years the public health has been experiencing legitimate threat due to COVID-19 virus infection. This article represents a holistic machine learning approach to get an insight of social media sentiment analysis on third booster dosage for COVID-19 vaccination across the globe. Here in this work, researchers have considered Twitter responses of people to perform the sentiment analysis. Large number of tweets on social media require multiple terabyte sized database. The machine learned algorithm-based sentiment analysis can actually be performed by retrieving millions of twitter responses from users on daily basis. Comments regarding any news or any trending product launch may be ascertained well in twitter information. Our aim is to analyze the user tweet responses on third booster dosage for COVID-19 vaccination. In this sentiment analysis, the user sentiment responses are firstly categorized into positive sentiment, negative sentiment, and neutral sentiment. A performance study is performed to quickly locate the application and based on their sentiment score the application can distinguish the positive sentiment, negative sentiment and neutral sentiment-based tweet responses once clustered with various dictionaries and establish a powerful support on the prediction. This paper surveys the polarity activity exploitation using various machine learning algorithms viz. Naïve Bayes (NB), K- Nearest Neighbors (KNN), Recurrent Neural Networks (RNN), and Valence Aware wordbook and sEntiment thinker (VADER) on the third booster dosage for COVID-19 vaccination. The VADER sentiment analysis predicts 97% accuracy, 92% precision, and 95% recall compared to other existing machine learning models.

P. Ghosh
CSE Department, Narula Institute of Technology, Kolkata, West Bengal, India

R. Dutta (✉) · N. Agarwal
ITER, Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India
e-mail: ritamdutta1986@gmail.com

S. Chatterjee
CSE Department, IMPS College of Engineering and Technology, Malda, West Bengal, India

S. Mitra
CSE Department, University of Glasgow, University Ave, Glasgow G12 8QW, UK

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_14

Keywords Holistic approach · NB · KNN · RNN · VADER sentiment analysis · Tweeter response · Booster dosage

1 Introduction

Social websites which are different forms viz. blogs, icon and forum sharing, video sharing social networks, microblogs etc. have used several online social media viz. Facebook, Instagram, YouTube, Linked-in, Twitter. These websites and mobile applications share various people's response globally.

In these social sites, different individuals across the world can express their discussion, comments in various styles like text, image, video, emoji [1, 2]. Social media with huge source of knowledge can gather user opinion and various polls regarding the expression. Microblog has become the simplest familiar and therefore the source of various data [3]. Twitter is one the microblog service that enables users to share, reply within a short time frame as tweets [4]. It provides a fashionable supply of knowledge which are utilized in various scientific studies that are using sentiment analysis to extract and analysis data which are expressed as tweets on various topics like market, election, share-trade prediction.

Linguistic Inquiry and Word Count (LIWC) is one of the tools of text extraction [5, 6]. Most of these tolls require programming, here in our work we have used Valence Aware wordbook and sentiment thinker (VADER), which work on sentiment analysis of tweets on third booster dosage for COVID-19 vaccination among various countries.

2 Literature Survey

A thorough literature survey containing text mining and sentiment analysis approaches have been performed in this work. Gurkhe et al. [7] in their paper have projected twitter information which is collected and processed from numerous sources and removed the content which does not hold any polarity. Bouazizi et al. [8] have used a tool SENTA for sentiment analysis of the tweets and calculate score according to sentiment. Gautam et al. have used a review classification on tweets [9], where they have used primary algorithms viz. Naïve Bayes, Support Vector Machine (SVM), Maximum Entropy used from NLTK module of Python. Amolik et al. [10] have used twitter sentiment analysis on Hollywood movie industry and they have compared Naïve Bayes and SVM algorithm on accuracy classification. Mukherjee et al. [11] have used a hybrid sentiment analysis tool TwiSent where spell check and linguistic handler have already been defined. Davidov et al. [12] have introduced a supervised sentiment analysis technique on twitter information. Neethu et al. [13] have used machine learning technique on SVM, Naïve Bayes, Maximum entropy on MATLAB platform for classification data. A typical design structure of tweets on terrorism attack and their possible activities has been represented by Garg et al. [14]. Lots of tweets after the attack with #tag has been used on Naïve Bayes algorithm which was used on huge data and analysis has been performed for characteristics of the comments. Hasan et al. [15] have projected a hybrid approach where the tweets are followed by the #tag on political trend. Several Urdu tweets are translated to English for analysis, where the Naïve Bayes and SVM approaches are used to build a structure. Bhavsar et al. [16, 17] have designed and projected a sentiment analysis method on python platform and the data source were collected from Kaggle. Classifications on user's emotions on positivity and negativity were done for accuracy finding. Otaibi et al. [41] have structured a model both on supervised and unsupervised algorithm. They have used Twitter API for extracting 7000 tweets which are based on comments on McDonald and KFC quality. The analysis was performed on R programming language platform.

3 Sentiment Analysis on Twitter

This system primarily consists of the following stages. The stages square measure mentioned in Fig. 1.

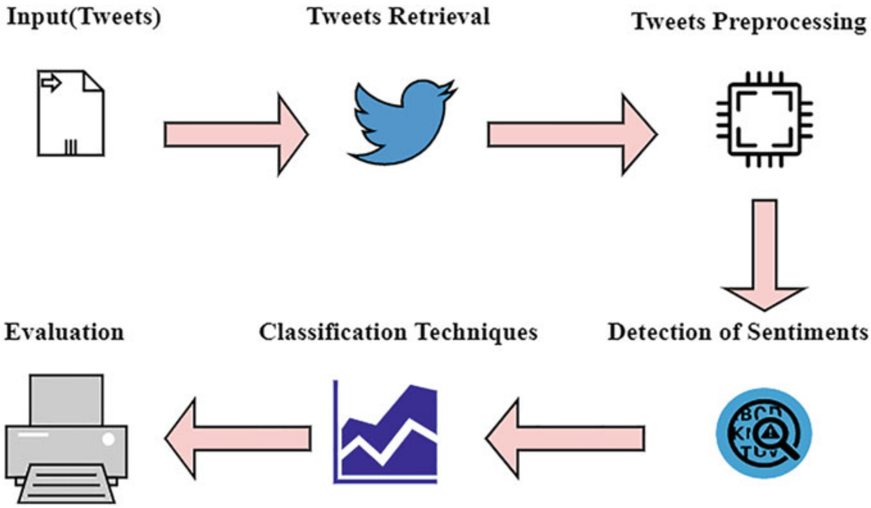


Fig. 1 Different stages of sentiment analysis

3.1 Data Extraction

For any machine learning model, database is the primary need. These data train themselves and predict the unseen data. Initially we opt for the topic with associated subject that will be gathered. The social media responses (tweets) are retrieved in unstructured, structured and semi structured form.

3.2 Data Pre-processing

In this step, within the collected social media responses (tweets) data pre-processing is performed. Here the large set of information is filtered by eliminating irrelevant, inconsistent, and yelling information. It functions by converting these datasets to lowercase and removing the duplicate values viz. punctuations, spaces, stops etc., further need to add contractions and lemmatizations.

3.3 Sentiment Detection

By incorporating data classification and data (tweet) mining the sentiment detection can be performed [18].

3.4 Sentiment Classification

Algorithmic ruled sentiment analysis is generally classified to two approaches viz. supervised learning and unattended learning. In the supervised learning, the Naïve Thomas Bayes, SVM and most entropy square measure accustomed execute the sentiment analysis [19–28].

3.5 Evaluation

The final output is analyzed to require call whether or not we must always prefer it or not [29–33]. In this step, within the collected social media responses (tweets) data pre-processing is performed. Here the large set of information is filtered by eliminating irrelevant, inconsistent, and yelling information.

4 Proposed Model Component Description

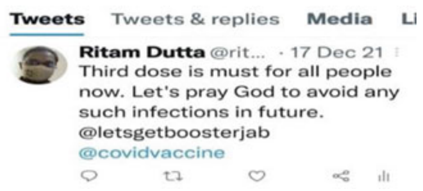
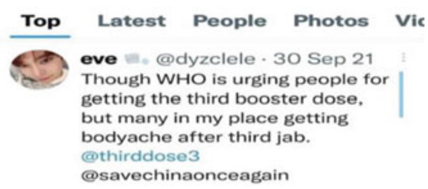
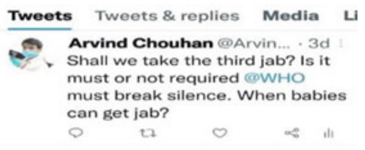
Twitter may be a social networking platform that enables its users to send and skim micro-blogs of up to 280-characters called "tweets". It allows registered users to browse and post their tweets through internet, short message service (SMS) and mobile applications. As a worldwide period of time communications platform, Twitter has quite four hundred million monthly guests and 255 million monthly active users round the world. Twitter's active cluster of registered members includes World leaders, major athletes, star performers, news organizations, and amusement retailers. It is presently accessible in additional than thirty-five languages.

Twitter was launched in 2006 by Jack Dorsey, Evan Williams, Biz Stone, and patriarch Glass. Twitter is headquartered in San Francisco, California, USA.

Here in Table 1, the sample dataset collected from social media (Twitter) has been showcased where the different sentiment categories have been identified by machine learning algorithms used in the model.

The Natural Language Toolkit (NLTK) has been used in our Valence Aware Dictionary for sEntiment Reasoning (VADER) model. NLTK provides free ASCII text file Python package that has many tools for building programs and classifying knowledge [34, 35]. The VADER model is a rule-based sentiment analysis tool that specifically

Table 1 Sample dataset collected from social media (Twitter)

Comments	Category
 Ritam Dutta @rit... · 17 Dec 21 Third dose is must for all people now. Let's pray God to avoid any such infections in future. @letsgetboosterjab @covidvaccine	Positive
 eve @dyzclele · 30 Sep 21 Though WHO is urging people for getting the third booster dose, but many in my place getting bodyache after third jab. @thirddose3 @savechinaonceagain	Negative
 Arvind Chouhan @Arvin... · 3d Shall we take the third jab? Is it must or not required @WHO must break silence. When babies can get jab?	Neutral

attuned to the user emotions expressed in social media [36, 37]. All instances of the info sets had accents and punctuation marks removed and then different data preprocessing techniques have been applied, viz. Lemma extraction [38], Stemming, Part of Speech (PoS) tagging [39, 40] and Summarization. Similar words of received twitter responses are identified using regular expressions and passed to a dictionary to label the data that can be used for supervised learning.

4.1 Flow Chart of Our Proposed Model

In this section, the flowchart of our proposed sentiment analysis model is depicted in Fig. 2. In the mentioned Algorithm 1 below, the steps to carry on the analysis process on the received texts are further shown in Fig. 3.

In our work, the VADER sentiment analysis tool has been used to get the simulated response. VADER is a lexicon and rule-based sentiment analysis tool which is specifically adjusted to the sentiments expressed in social media (twitter) and provides substantial results on texts from other domains.

4.2 Different Machine Learning Models Used for Performance Comparison

The machine learning models are trained with some data and are used to make predictions on unseen data. The specialty of machine learning models is that they can extract and learn the features of a dataset using some feature selection technique and therefore don't require human intervention. This section describes the machine learning models used in our proposed work.

4.2.1 Naïve Bayes

The Naïve Bayes classification is a well-known supervised machine learning approach that makes predictions based on some probability. It is based on the Bayes theorem to determine the probability value, calculated as shown below:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \tag{1}$$

where,

P(X|C): likelihood

P(C|X): posterior probability

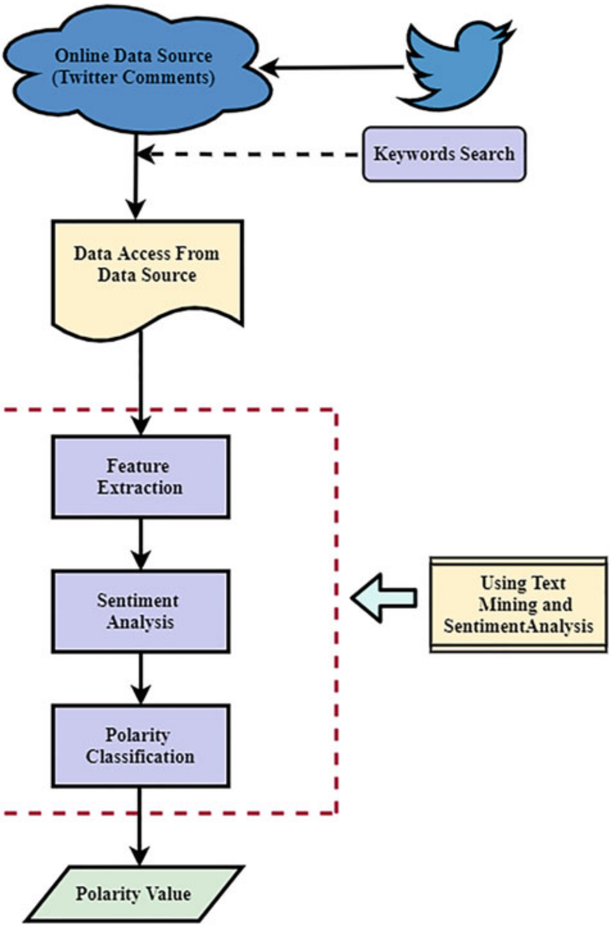


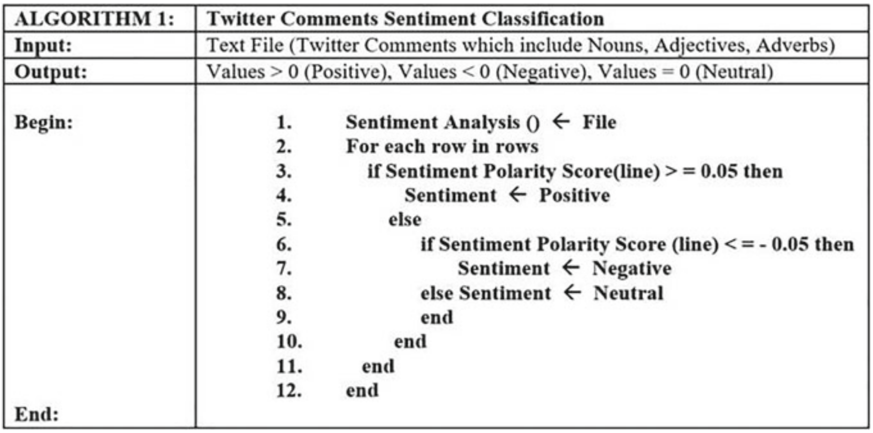
Fig. 2 Flowchart of our proposed sentiment analysis model

P(C): class probability

P(X): predictor probability

4.2.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor is the simplest yet widely used supervised Machine Learning algorithm. It is widely used in text mining, pattern recognition, and many other fields. It groups similar types of data with respect to k neighbors and based on the similarity, the classification is done. In our work, we have used the grid search technique to determine the "k" value and it was observed that k = 15 gave good results as shown in Fig. 4b.



ALGORITHM 1:	Twitter Comments Sentiment Classification
Input:	Text File (Twitter Comments which include Nouns, Adjectives, Adverbs)
Output:	Values > 0 (Positive), Values < 0 (Negative), Values = 0 (Neutral)

```
Begin:
    Sentiment Analysis () ← File
    For each row in rows
      if Sentiment Polarity Score(line) > = 0.05 then
          Sentiment ← Positive
        else
            if Sentiment Polarity Score (line) < = - 0.05 then
                Sentiment ← Negative
            else Sentiment ← Neutral
            end
          end
      end
    end
End:
```

Fig. 3 Proposed algorithm for twitter comments sentiment classification

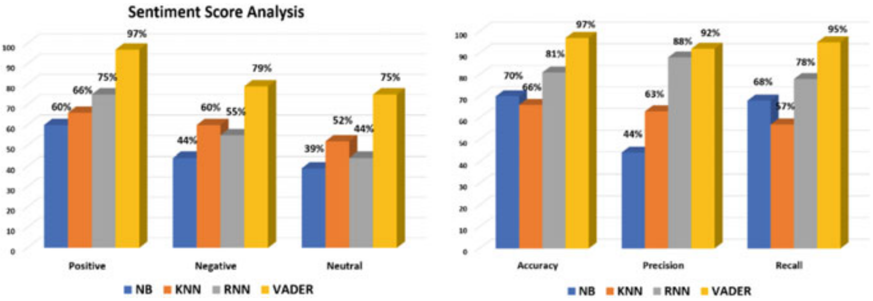


Fig. 4 **a** Sentiment score analysis using NB, KNN, RNN and VADER algorithms, **b** Accuracy, Precision and Recall for different sentiment analyzer models

4.2.3 Recurrent Neural Network (RNN)

Recurrent Neural Networks are a type of Neural Network that remember old data to make future predictions. They analyze the data more efficiently as compared to other machine learning models as the latter uses the current data only to make future predictions. The RNN model has a memory that remembers the relevant past information and forgets the irrelevant information. As the same parameters are used throughout the layers, the complexity of the model is reduced to a large extent. The RNN model is widely used in text mining, natural language processing, and many other tasks. The parameters used in our RNN model are shown in Table 2.

Table 2 Parameters used in our RNN model

Parameters	RNN
Layers used	4
Fully connected layers used	2
Number of nodes in each layer	128, 64, 32, 16
Number of nodes in fully connected layers	16, 1
Optimization technique	RMSprop
Activation function	SoftMax
Epochs	100
Learning rate	0.0001

4.2.4 VADER

The Valence Aware Dictionary and sEntiment Reasoner sentiment analysis tool being an unsupervised learning approach, which is able to detect the polarity of the sentiment (positive, negative, or neutral) of a given text when the data is analyzed as unlabeled. Therefore, our proposed VADER model is less expensive compared to other existing supervised learning approaches. Orthodox sentiment analyzer models are given opportunity to learn from labeled training data, which complexes the process. The VADER sentiment analyzer is smart to get the job done without the label formation. VADER uses a lexicon of sentiment-related words to determine the overall sentiment of a given body of text.

4.3 Evaluation Matrix

This section describes the metrics used to evaluate the performance of our models. The three evaluation metrics incorporated in our work are described below:

Accuracy: It represents the total predictions correctly made.

Precision: It is the ratio of the number of correct positive results divided by the number of positive results predicted by the classifier.

Recall: It is the ratio of the number of correct positive results divided by the number of relevant samples.

5 Simulation Results and Discussion

For the result analysis, we have continued by closing computation supported information which were collected from social media (twitter) information. The sentiment analysis tool incorporates word-order sensitive relationships between terms and

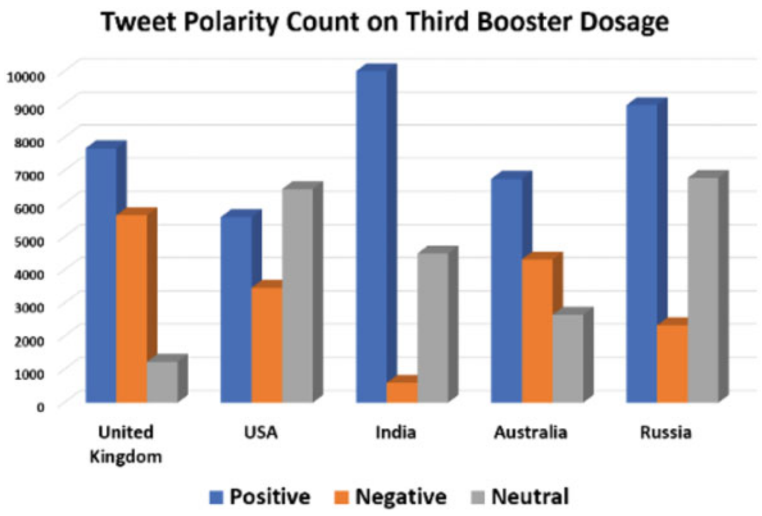


Fig. 5 Twitter polarity count on third booster dosage using VADER sentiment Analysis

then collected information are processed to identify positive, negative, and neutral sentiments.

Here in Fig. 4a, a bar chart comparison on sentiment score analysis fetched from social media (twitter) responses modeled by four machine learning algorithms i.e., NB, KNN, RNN and VADER is performed. The performance of these models is evaluated using the well-known machine learning evaluation metrics viz. accuracy, precision, and recall as shown in Fig. 4b. The tweeter responses on the usage and significance of third booster dosage for COVID-19 vaccination have been categorically modeled using four machine learning algorithms, where the VADER sentiment analyzer shows best accurate results compared to others.

The twitter polarity count on third booster dosage has also been compared between top five superpower countries using VADER sentiment analysis as shown in Fig. 5.

6 Conclusion and Future Scope

In our paper on the concurrent result of analyzing twitter responses of various countries on usage and probable significance of third booster dosage to combat COVID-19 infection is recorded. The World Health Organization wishes to speculate into the medicine market. The machine learning models viz. NB, KNN, RNN, and VADER were used to analyze the sentiments of the humans portrayed in tweets. This comparative study has proved the results where VADER sentiment analysis was performed with 97% accuracy, 92% precision, and 95% recall. The results have shown a robust co-relation between Twitter comments in sentiment polarity. The VADER being an unsupervised learning approach has proposed a model that is less expensive compared to other existing supervised learning approaches. This proposed approach can be used for other contagious infections if needed in future.

References

1. Jansen BJ, Hang MZ, Sobeland K, Chowdury A (2009) Twitter power: Tweets as electronic word of mouth. J Am Soc Inf Sci Technol 60(11):2169–2188
2. Kharde V, Sonawane P (2016) Sentiment analysis of twitter data: a survey of techniques. Int J Comput Appl. ArXiv1601.06971
3. Selvaperumal P, Suruliandi A (2014) A short message classification algorithm for tweet classification. Int Conf Recent Trends Inf Technol 1–3
4. Singh T, Kumari M (2016) Role of text pre-processing in twitter sentiment analysis. Procedia Comput Sci 89:549–554
5. Tausczik YR, Pennebaker JW (2010) The psychological meaning of words: LIWC and computerized text analysis methods. J Lang Soc Psychol 29(1):24–54
6. Gilbert CJ (2016) Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Eighth international conference on weblogs and social media (ICWSM-14)
7. Gurkhe D, Pal N, Bhatia R (2014) Effective sentiment analysis of social media datasets using Naïve Bayesian classification. Int J Comput Appl
8. Bouazizi M, Ohtsuki T (2018) Multi-class sentiment analysis in Twitter: what if classification is not the answer. IEEE Access. 6:64486–64502
9. Gautam G, Yadav D (2014) Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In: 7th international conference on contemporary computing
10. Amolik A, Jivane N, Bhandari JM, Venkatesan M (2016) Twitter sentiment analysis of movie reviews using machine learning techniques. Int J Eng Technol 7(6):1–7
11. Mukherjee S., Malu A, Balamurali AR, Bhattacharyya P (2013) TwiSent: a multistage system for Analyzing sentiment in Twitter. In: Proceedings of the 21st ACM international conference on information and knowledge management
12. Davidov D, Sur O, Rappoport A (2010) Enhanced sentiment learning using Twitter hashtags and smileys. In: Proceedings of the 23rd international conference on computational linguistics, posters
13. Neethu M, Rajasree R (2013) Sentiment analysis in twitter using machine learning techniques. In: 4th international conference on computing, communications and networking technologies, IEEE, Tiruchengode, India
14. Garg P, Garg H, Ranga V (2017) Sentiment analysis of the Uri terror attack using Twitter. In: International conference on computing, communication and automation, IEEE, Greater Noida, India
15. Hasan A, Moin S, Karim A, Shamshirb S (2018) Machine learning based sentiment analysis for Twitter accounts, Licensee, MDPI, Switzerland
16. Bhavsar H, Manglani R (2019) Sentiment analysis of Twitter data using python. Int Res J Eng Technol 3(2):41–45
17. Sirsat S, Rao S, Wukkadada B (2019) Sentiment analysis on Twitter data for product evaluation. IOSR J Eng 5(1):22–25
18. Behdenna S, Barigou F, Belalem G (2018) Document level sentiment analysis: a survey. In: EAI endorsed transactions on context-aware systems and applications
19. Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. Comput Linguist J 267–307
20. Tong RM (2001) An operational system for detecting and tracking opinions in on-line discussions. In: Working notes of the SIGIR workshop on operational text classification, pp 1–6

21. Turney P, Littman M (2003) Measuring praise and criticism: inference of semantic orientation from association. ACM Trans Inform Syst J 21(4):315–346
22. Kaur L (2016) Review paper on Twitter sentiment analysis techniques. Int J Res Appl Sci Eng Technol 4(1)
23. Miller GA, Beckwith R, Fellbaum C, Gross D, Miller KJ (1990) Introduction to WordNet: an on-line lexical database. Int J Lexicogr 3(4):235–244
24. Mohammad S, Dunne C, Dorr B (2009) Generating high-coverage semantic orientation lexicons from overly marked words and a thesaurus, In: Proceedings of the conference on empirical methods in natural language processing
25. Harb A, Plantie M, Dray G, Roche M, Trousset F, Poncelet F (2008) Web opinion mining: How to extract opinions from blogs? In: Proceedings of the 5th international conference on soft computing as transdisciplinary science and technology (CSTST 08), pp 211–217
26. Turney PD (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics, pp 417–424
27. Wang G, Araki K (2007) Modifying SOPMI for Japanese weblog opinion mining by using a balancing factor and detecting neutral expressions. In: Human language technologies: the conference of the North American chapter of the association for computational linguistics, companion volume, pp 189–192
28. Rice DR, Zorn C (2013) Corpus-based dictionaries for sentiment analysis of specialized vocabularies. In: Proceedings of NDATAD, pp 98–115
29. Su KY, Chiang TH, Chang JS (1996) An overview of corpus-based statistics oriented (CBSO) Techniques for natural language processing. Comput Linguist Chin Lang Process 1(1):101–157
30. Su KY, Chiang TH (1990) Some key issues in designing MT systems. Mach Transl 5(4):265–300
31. Su KY, Chiang TH (1992) Why corpus-based statistics oriented machine translation. In: Proceedings of 4th international conference on theoretical and methodological issue in machine translation, Montreal, Canada, pp 249–262
32. Su KY, Chang JS, Una Hsu YL (1995) A corpus-based two-way design for parameterized MT systems: rationale, architecture and training issues. In: Proceedings of the 6th international conference on theoretical and methodological issues in machine translation, TMI-95, pp 334–353
33. Moss HE, Ostrin RK, Tyler LK, Marslen WD (1995) Accessing different types of lexical se-mantic information: evidence from priming. J Exp Psychol Learn Mem Cogn 21(1):863–883
34. Natural Language Toolkit. http://www.nltk.org/. Last Accessed 20 Nov 2018
35. Bird S, Loper E, Klein E (2009) Natural language processing with python. O'Reilly Media Inc.
36. Gilbert CJ (2014) Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: 8th international confernece on weblogs and social media
37. Elbagir S, Yang J (2019) Twitter sentiment analysis using natural language toolkit and VADER sentiment. LNCS 232:342–347
38. Natural Language Processing with neural networks. http://nilc.icmc.usp.br/nlpnet/. Last Accessed 05 Dec 2021
39. Fonseca ER, Rosa JLG (2013) A two-step convolutional neural network approach for semantic role labelling. In: Proceedings of the 2013 international joint conference on neural networks-2013, pp 2955–2961
40. Fonseca ER, Rosa JLG (2013) Mac-Morpho revisited: towards robust part-of-speech tagging. In: Proceedings of the 9th Brazilian symposium in information and human language technology, pp 98–107
41. Rahman SA, Al Otaibi FA, AlShehri WA (2019) Sentiment analysis of Twitter data. In: 2019 international conference on computer and information sciences, pp 1–4. https://doi.org/10.1109/ICCISci.2019.8716464

Non-rigid Registration of De-noised Ultrasound Breast Tumors in Image Guided Breast-Conserving Surgery

Sanjib Saha

Abstract It's an article based on medical ultrasound-to-ultrasound non-rigid registration of breast tumors in image-guided surgery. Firstly, the paper discusses the challenges behind ultrasound image registration. Secondly, the paper establishes scan conversion using Lee filters along radial as well as horizontal directions depending on the minimum mean square error (MMSE) is a reasonable choice to convert signal-dependent or pure multiplicative noise to an additive one and that produces speckle free ultrasound image. This paper also introduces unique aspects of the registration framework for non-rigid (elastic) deformations and presents a novel non-rigid registration associated with a unique basis function where each local control point strikes the deformed structure of the curve over the range of criterion values. Piecewise cubic polynomial form splines (B-splines) are used to get the distortion field among two ultrasound images and one of the popular similarity measure criteria on sum of squared difference (SSD) is used to find the dis-similarity among mono-modal images. The line search approach of the Quasi-Newton Limited-Memory (LM) Broyden–Fletcher–Goldfarb-Shanno (BFGS) algorithm is used to optimize the dissimilarity errors. Thirdly, this proposed work for non-rigid registration is applied in breast-conserving surgery of breast tumors between pre-operative and intra-operative ultrasound. This novel approach is computationally efficient.

Keywords Non-rigid registration · Ultrasound image registration · Breast tumor · Image guided surgery

1 Introduction

Image registration aligns two images i.e., the reference image and the test image. The images taken by various sensors or at the various time or from various views are to be compared when a small sub-portion of the image needs to be searched in

S. Saha (✉)
Department of Computer Science and Engineering, Dr. B. C. Roy Engineering College, Durgapur, India
e-mail: sanjib.saha@bcrec.ac.in

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_15

another image, then these images need to align properly into a common coordinate system to identify the differences and the image with the sub-portion is found. This is useful in medical treatments. Medical images of a similar scenario are recorded by various sensors that are used for correct clinical diagnosis. For this, medical images should be aligned properly for clear monitoring. Medical ultrasound is one of the important imaging modalities which provides a quantitative way of solving medical problems and uses it on humans for scientific work. Ultrasound is non-invasive, non-ionizing, cheaper, and real-time that's why it is useful for all types of patients. Ultrasound-to-ultrasound image registration [1, 2] procedure is a challenging task for having noise, poorly defined image gradients, artifacts, and low image contrast resolution which makes it difficult to achieve accuracy in the alignment of images. By the presence of a signal-dependent or pure multiplicative noise [3] known as speckle, the usefulness of ultrasound imaging is degraded. The filtering scan conversion is to convert multiplicative noise to an additive one. Non-rigid transformation [4] function allows the test image to be disfigured to compare with the reference image. The alignment of pre-operative actual images and intra-operative deformed images helps the surgeon in proper guidance at the time of surgery. Medical images captured at various time from various angles that guide in monitoring the disease over time can be inferred. This medical image registration can become quite helpful for IGS [5–8] as it can cure the tumor, and cancer with the best optimization result. The necessity of better registration techniques can be beneficial for surgeons in IGS with shorter procedures in the surgical field and improved hand–eye coordination. Visualization of interesting anatomical areas in medical images is a vital requirement for the IGS system which is an alternative to conventional surgery. The other parts of the article are arranged as mentioned: literature survey is discussed in Sect. 2. Section 3 describes background technologies. Section 4 introduces proposed method. Section 5 discusses results of the proposed method. Ultimately, Sect. 6 concludes the paper.

2 Literature Survey

In medical science, the ultrasound breast image registration [9–11] in IGS proves the least fault. Breast cancer is one of the most common diseases related to cancer which is the greatest cause of cancer-related deaths among women all over the world. In the USA, it is the second most frequently diagnosed cancer after lung cancer, and 1 out of 8 among women lifetime is at risk of diagnosing Breast Cancer. Breast cancer and lung cancer are the most common cancer diseases in India now. West Bengal, Gujarat, Bihar, Odisha, Kerala, Delhi, Tamil Nadu, Maharashtra, Karnataka, Rajasthan, Madhya Pradesh, and Chhattisgarh are the states of India where several women are suffering from breast cancer. In West Bengal, about 146 out of 100,000 women are suffering from breast cancer. Breast cancer is a complicated disease with mono-modal or multi-modal approaches for treatment. Surgery is the initial treatment of primary breast tumors. Breast surgical options include lumpectomy (removal of the tumor) and mastectomy (removal of the breast). Lumpectomy is also known as

Breast-Conserving Surgery (BCS). The non-rigid registration between pre-operative actual ultrasound and intra-operative deformed ultrasound can reduce re-excision rates in breast-conserving surgery [12] which is helpful to the multi-disciplinary treatment of breast cancer victims. Medical image registration can be divided into the following process. The distinct features (edges, intensities, constructs) of each image are to be detected. Different similarity measurement is carried out to get the correlation between the features of the reference image and the test image. By using feature comparison, the parameters for the transformation of the test image to the reference image are estimated. This mapping function obtained from the transform model estimation uses appropriate interpolation and optimization techniques to get a better and faster transformation.

3 Background Technologies

3.1 Image Registration Using Transformation Functions

Image registration geometrically aligns two images of a similar scene captured from similar or dissimilar modalities. Image Registration is the method of plotting points from one image to respective points in a different image. In the article, the reference image is indicated by I_r, whereas the test image is indicated by I_t. The purpose of registration is to approximate the optimal transformation T in the image domain Ω of two images I_r and I_t. The registration optimizes the energy of the form in Eq. (1):

$$Tr(I_r, I_t.T) + Rg(T) \tag{1}$$

This objective function in Eq. (1) involves 2 terms. The 1st term, T_r, computes the level of arrangement among a reference image I_r and a test image I_t. The optimization can be achieved by either minimizing or maximizing the objective function and how the (dis) similarity criterion is selected. The transformation $T(x)$ at every point $x \in \Omega$ is considered as the sum of an identity transformation along with the displacement field $v(x)$ is written in Eq. (2):

$$T(x) = x + v(x) \tag{2}$$

The second term, R_g, regularizes or normalizes the transformation to get the required solution by overcoming the difficulty associated with it. Hence, image registration includes the following steps: deformation or transformation model, (dis) similarity measure using objective function, regularization, and optimization.

This registration can be rigid (non-elastic) and non-rigid (elastic) [13]. The rigid registrations are translation and rotation, and preserves length and angle after transformation. Translation, rotation, scaling, and shearing are used as an affine transformation, and it could not preserve lengths and angles but must preserve parallel lines.

In 1D, the affine transformation is parameterized by 4 degrees of freedom (translation, rotation, scaling, and shearing) and 12 degrees of freedom in 3D. But rigid transformation is restricted by 6 degrees of freedom (translation, rotation) in 3D. Non-rigid registrations make use of non-rigid transformation functions that involve a large number of parameters. 2D affine transformation T_A using the transformation order shearing, scaling, rotation and at last translation, a point or location (or an object) can be written as follows in Eqs. (3) and (4) (by adding homogeneous coordinate $z = 1$ to each location):

$$T_A(x, y) = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = Translation\ Rotation\ Scaling\ Shearing \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

Therefore, the 2D affine transformation is parameterized by sharing in *x*–*y* direction (sh_x, sh_y), scaling in *x*–*y* direction (s_x, s_y), clockwise rotation in *x*–*y* direction (sinθ, cosθ), and translation in *x*–*y* direction (t_x, t_y) i.e., maximum 8 degrees of freedom.

$$\begin{aligned}
\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & sh_x \\ 0 & 1 & sh_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x + sh_x \\ y + sh_y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (x + sh_x)s_x \\ (y + sh_y)s_y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \{(x + sh_x)s_x\}\cos\theta + \{(y + sh_y)s_y\}\sin\theta \\ -\{(x + sh_x)s_x\}\sin\theta + \{(y + sh_y)s_y\}\cos\theta \\ 1 \end{bmatrix} T_A(x, y) = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} [\{(x + sh_x)s_x\}\cos\theta + \{(y + sh_y)s_y\}\sin\theta] + t_x \\ [-\{(x + sh_x)s_x\}\sin\theta + \{(y + sh_y)s_y\}\cos\theta] + t_y \\ 1 \end{bmatrix}
\end{aligned} \tag{4}$$

3.2 Image De-noising Using Lee Filters

Speckle noise degrades the usefulness of ultrasound images. As speckle noise is the multiplicative and non-white process, and the filtering scan conversion is a reasonable choice to convert signal-dependent or pure multiplicative noise to an additive one. To evaluate the consistent speckle in medical ultrasound images and corresponding to the model is as in Eq. (5):

$$I^{'}(x) = i(x) + f(x) * g(x) \tag{5}$$

where $I'(x)$: the real noise image, $i(x)$: the unobservable actual image, $f(x)$ and $g(x)$ are the points spreading function of the speckle and the Gaussian noise, respectively.

Mean Square Error (MSE) is to assess the speckle reduction in the case of multiplicative noise by finding the total amount of differences between the actual images $I'(x_i)$ and the de-noised images $I(x_i)$. Lower MSE values display that the filtering effect is better, and filtered image quality is much greater. MSE is calculated as given in Eq. (6):

$$MSE = \frac{1}{n}\sum_{i=0}^{n}[I'(x_i) - I(x_i)]^2 \tag{6}$$

The scan conversion using Lee filters along the radial and horizontal direction depends on the minimum mean square error (MMSE), which is producing a speckle free image managed by the below Eq. (7):

$$I(x) = i(x)F(x) + i^{'}(x)(1 - F(x)) \tag{7}$$

where i'' indicates the mean value of the intensity between the filter kernel, and $F(x)$ is the adaptive filter coefficient taken out by the following Eq. (8):

$$F(x) = 1 - \frac{c_n^2}{c_{ni}^2 - c_n^2} \tag{8}$$

where c_{ni} indicates the coefficient of variation of the noised image and c_n is the coefficient of variation of the noise.

Alternatively, filtering scan conversion [14] can be performed by spatial linear adaptive and nonlinear filters along the radial direction and horizontal direction using the Lee and Kuan filters, respectively.

3.3 Similarity Measure Using Sum of Squared Difference (SSD)

The addition of squared differences of bias and scale normalized intensities in every image is considered for calculating mismatch among these images. There are many ways to measure insensitive to image contrast, The 1st method is to divide the mean-normalized intensities in every image by the standard deviation of the intensities and another method normalizes the measurement according to image contrast. The SSD cost function $C(v)$ according to the transformation $T(x)$ with the displacement field $v(x)$ on the test image $I_t(x)$ and the given reference image $I_r(x)$ is calculated as follows in Eqs. (9) and (10), and shown in Fig. 1:

$$\begin{aligned} C(v) &= \sum_{i=0}^{n} [I_r(x_i) - I_t(T(x_i))]^2 \\ &= \sum_{i=0}^{n} [I_r(x_i) - I_t(x_i + v(x_i))]^2 \\ &= \sum_{i=0}^{n} [I_r(x_i) - I_t(x_i) - vs]^2 \\ &= \sum_{i=0}^{n} [e - vs]^2 \end{aligned} \tag{9}$$

$$\frac{dC}{dv} = \frac{d}{dv}[e - vs]^2$$

$$\frac{dC}{dv} = \frac{d}{dv}[e^2 - 2evs + v^2 s^2]$$

$$0 = 0 - 2es + 2vs^2 \left(if \frac{dC}{dv} = 0 \right)$$

$$v = \frac{e}{s}$$

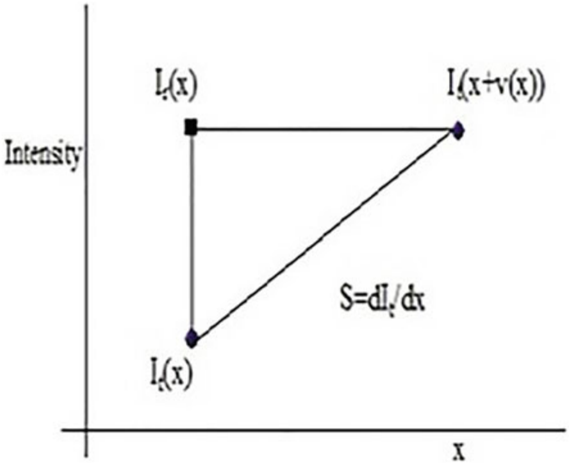


Fig. 1 SSD cost function for $I_r(x)$, $I_t(x)$ and $T(x)$

$$
\begin{aligned}
v(x) &= \frac{I_r(x) - I_t(x)}{\frac{dI_t(x)}{dx}} \\
&= \frac{I_r(x) - I_t(x)}{\nabla I_t(x)}
\end{aligned} \tag{10}
$$

3.4 Non-rigid Transformation Using B-Splines

The transformation is a combination of global transformation (or affine) and local transformation (or splines).

$$T(x) = T_{global}(x) + T_{local}(x)$$

The global transformation model relates to the universal motion of the objects. This model cannot trade with distortion of the image to be recorded. To perfectly register the deformation in the image by operating a fundamental network of control points, the Free Form Deformation (FFD) model using B-spline [15, 16] is popular. Some other spline methods are globally controlled; altering in control point at one location, in turn, effecting the location of the most control points. Whereas, B-splines are locally controlled, change in the control point effects the transformation only in the local region of the control point. The control points act as factors of the B-Spline FFD. The degree of freedom for non-rigid deformation can be tuned by changing the resolution of the network of control points. Huge gapping of control points tends to be the modeling of global non-rigid transformation. With decreasing in the spacing of the control points, and transformation permits to modeling of local non-rigid deformation. The term Penalty is appended to the cost function to smooth the spline-based FFD transformation. B-spline is a simplification of the Bezier curve. To increase or decrease the order in the Bezier curve we need to increase or decrease the number of polygon vertices and there are only global control points but no local control. B-spline is a spline function that has minimal support according to a particular degree, smoothness, and domain of partition. The B-spline function is a union of flexible bands that crosses the number of points that are known as local control points and create smooth curves. To alter the appearance of a B-spline curve, one can alter more than one of these control factors: the degree of the curve (k), the locations of control points (P), and the locations of knots (t). The no. of control points is $n + 1$ and the curve is made of $n\text{-}k + 2$ segments. Each segment is influenced by k points.

The FFD using 1-D cubic B-splines (B) and displacement field v can be written as Eqs. (11) and (12), and shown in Fig. 2:

$$T_S(x) = x + v(x)$$

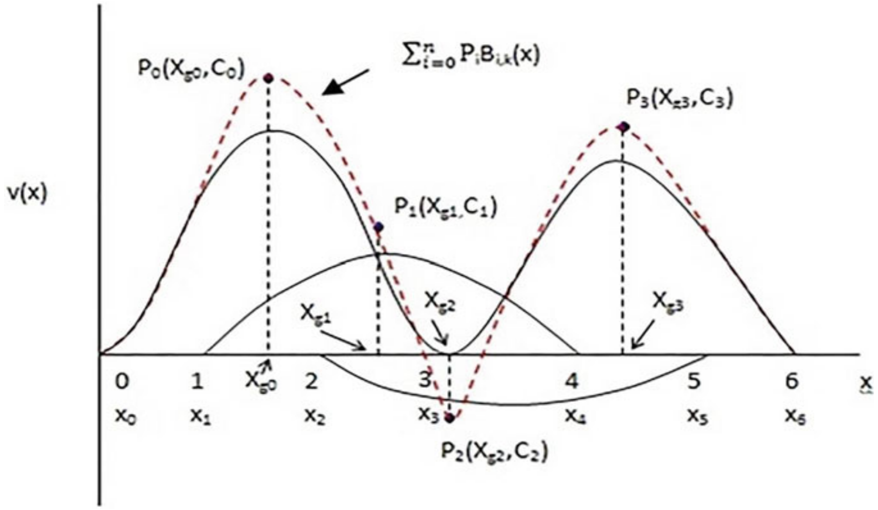


Fig. 2 B-spline where each local control point affects the deformed structure of the curve over the range of factors values

$$v(x) = \sum_{i=0}^{n} P_i B_{i,k}(x)\,(where\,0 \le x \le n - k + 2) \tag{11}$$

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(x) \tag{12}$$

$$(t_i)_{i=0}^{n+k}\, is\, the\, knot\, values\,(0 \le i \le n + k)$$

$$t_i = 0\,(if\, i < k)$$

$$t_i = i - k + 1\,(if\, k \le i \le n)$$

$$t_i = n - k + 2\,(if\, i > n)$$

$$\begin{aligned} B_{i,0}(x) &= 1(if\, t_i \le x < t_{i+1}) \\ &= 0\,(otherwise) \end{aligned}$$

For example (Fig. 2), the order of the red curve k = 2 and number of control points n + 1 = 4 (i.e., P_0, P_1, P_2, P_3), so, n = 3 as given in Eqs. 13 and 14. Therefore, the curve is made of n−k + 2 = 3 segments. Each segment is influenced by k points. Each control point is correlated with a unique basis function. Each point influences the structure of the curve over a range of parameter values.

$$v(x) = \sum_{i=0}^{3} P_i B_{i,k}(x)\ (where\, 0 \leq x \leq n - k + 2) \tag{13}$$

$$v(x) = P_0 B_{0,2}(x) + P_1 B_{1,2}(x) + P_2 B_{2,2}(x) + P_3 B_{3,2}(x) \tag{14}$$

$$B_{i,k}(x) = \frac{x - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(x) + \frac{t_{i+k} - x}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(x)$$

$$B_{0,2}(x) = \frac{x - t_0}{t_1 - t_0} B_{0,1}(x) + \frac{t_2 - x}{t_2 - t_1} B_{1,1}(x)$$

$$B_{1,2}(x) = \frac{x - t_1}{t_2 - t_1} B_{1,1}(x) + \frac{t_3 - x}{t_3 - t_2} B_{2,1}(x)$$

$$B_{2,2}(x) = \frac{x - t_2}{t_3 - t_2} B_{2,1}(x) + \frac{t_4 - x}{t_4 - t_3} B_{3,1}(x)$$

$$B_{3,2}(x) = \frac{x - t_3}{t_4 - t_3} B_{3,1}(x) + \frac{t_5 - x}{t_5 - t_4} B_{4,1}(x)$$

$$B_{0,1}(x) = \frac{x - t_0}{t_0 - t_0} B_{0,0}(x) + \frac{t_1 - x}{t_1 - t_1} B_{1,0}(x)$$

$(t_i)_{i=0}^{5}\, is\, the\, knot\, values$ $(0 \leq i \leq n + k)$ i.e., 0 < = i <= 5.
Therefore, t_i = {0,0,1,1,3,3} as
t_0 = 0, t_1 = 0 (if i < k i.e. 0,1 < 2)
t_2 = i−k + 1 = 1, t_3 = 1 (if k <= i <= n i.e. 2 <= 2, 3 <= 3)
t_4 = n−k + 2 = 3, t_5 = 3 (if i > n i.e. 4, 5 > 3)

3.5 Optimization Using Line Search Based Limited-Memory BFGS

The line search approach finds the direction of the objective function along which the function is simplified, then it works out the step size to determine how much it can do more along the given direction. Gradient descent, Newton's method, and Quasi-Newton's methods are used in computing the descent direction. Quasi-Newton methods are used to find the maxima and minima of objective functions. Limited-memory BFGS [17] is an optimization algorithm in the family of Quasi-Newton methods that estimates the Boyden–Flecter–Glodfarb–Shanno (BFGS) algorithm with the help of a finite quantity of System Storage. This method is particularly applicable for problems consisting of a huge number of conditions and stores only very less no. of vectors from which to estimate the current values. As a result, linear memory need is sufficient for LM-BFGS optimization. It relates to one specific application that is known to be quite fast and has performed several iterations of optimization. LM-BFGS method is specifically convenient for optimization complications with many conditions. The replacement used in LM-BFGS is to use only an estimation of the true Hessian (matrix which organizes all the second partial derivatives of a function used in optimizing multivariable functions), and to build this approximation up iteratively. The main LM-BFGS algorithm executes one–one iteration instead of using any Hessian information and then starts the iteration, using the previous iteration's slope.

4 Proposed Method

4.1 Proposed Framework

(1) First, to make speckle-free medical ultrasound images using Lee filters scan conversion in radial and horizontal directions is based on the minimum mean square error.
(2) The second, is to find out the space transformation of the reference image and the test image using piecewise cubic polynomial basis splines.
(3) The third, for measuring the similarity degree of the reference image and test image is using the sum of squared difference; and
(4) The fourth, to perform the similarity measure that reaches the optimal value (parameter optimization) using the optimization algorithm in the family of Quasi-Newton methods that estimate the BFGS algorithm with the help of a limited amount of system memory (Fig. 3).

Here the proposed non-rigid image registration approach can register the two 2D images of the same modality and different sizes. The piecewise cubic polynomial basis spline grid-based local transformation is used to find the deformation/displacement field to get the deformed image from the reference image and the test image. The image that will be recorded is the test image and the reference or instance image is the image on which the test image will be recorded. The image dissimilarity measure between the reference image and test/deformed image is calculated using one of the popular metrics i.e., sum of squared difference (SSD). The B-spline control grid is defined or initialized with a uniform 2D grid B-spline knot spacing in X and Y directions that can be used to calculate maximum refinement and to transform the test image using the reference image, instead of a central gradient where the forward transformation field of the pixels in X and Y direction was seen from the test image to the reference image. The matrix form of transformation output is converted to a vector for reshaping. The test image which is transformed is multiplied by the individual pixel error before calculation of the total (mean) similarity error. One of the important phases like optimization is done using the limited memory BFGS steepest method. The calculation of derivatives and registration value and gradient of two images is done using the delta and forward gradient. Interpolation is

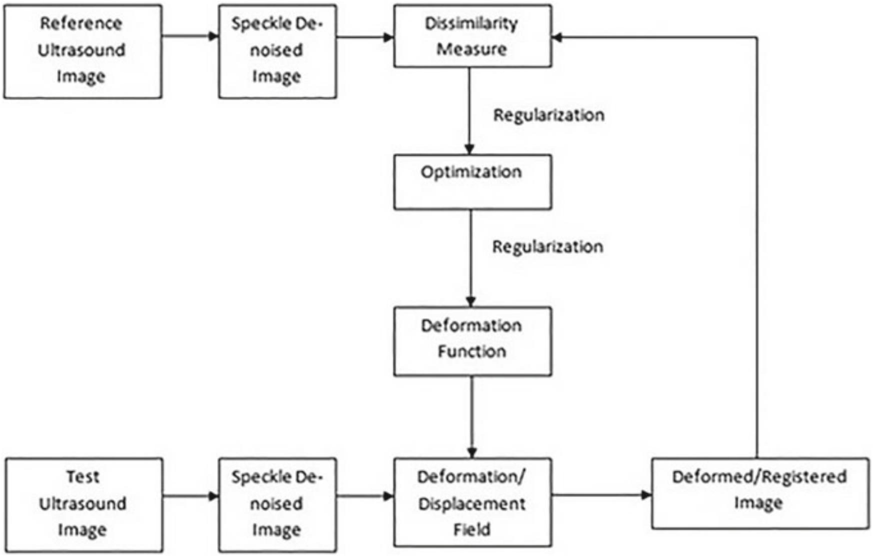


Fig. 3 Block diagram of proposed framework

one of the important steps. The cubic interpolation is used to transform a 2D image in a backward way. In the basis spline, each interval has different functions that connect all those points to get a spline function. Splines are used because at each interval we get an approximate function. The approximate function for example when there is jumble data in higher-order polynomial. Spline provides a superior approximation of the behaviour of functions that has local changes. A thin flexible strip connects the number of points. This strip or spline is also called a knot. To understand spline, let us draw different lines between the intervals. Now if we want to connect them in such a way that they are continuous, then they are called the first order spline, if we connect them in such a way that they are continuous as well as differentially continuous we call them second-order spline. If we connect them in such a way that they are first-order continuous and second-order also continuous, indicating their continuity is maintained and slopes are equal, and curvatures are also equal is called a cubic spline.

Two images piece wise, each using a linear transformation is registered in a piecewise cubic (PC) transformation. The regions may differ in shape and sizes—in theory. Only triangular regions have been used so far. Although the PC plotting is an uninterrupted process, it is not so smooth. With the area being less or with small geometric differences between images, the PC may the sufficient. The respective triangles in the test image are found to be triangulating the control points in the instance image by studying the respective control points in the test image. The registration accuracy will be influenced by the choice of triangulation. Polynomials of higher degrees are adjusted to the triangles with the coefficients of the polynomials determined in such a way that the polynomial slopes at two sides of a triangle edge become the same.

This is to make tangent continuity across triangles. Measures to fit smooth surfaces piecewise to triangular meshes are suggested. These piecewise smooth surfaces can be considered as the unit of a transformation while recording images with local geometric differences. Thus, these can be considered as a unit of transformation. These keep a local deformation or inexactitude among the control points. Thus, these are very apt in being used as the unit of transformation function for the registration images with local geometric differences. For the purpose of non-rigid registration, PC transformation is equipped with efficiency and sufficiency frequently. Piecewise cubic functions are also utilized in non-rigid registration. Image regions within the convex hull of the control points are recorded using piecewise techniques.

4.2 Application of Proposed Method in Image Guided Breast Tumor Surgery

The medical image registration can become quite helpful for IGS as it can cure the tumor, and cancer with the best optimization result. The alignment of pre-operative actual images and intra-operative deformed images greatly helps the surgeon in appropriate guidance at the time of surgery. The necessity of better registration techniques can be beneficial for surgeons in IGS with shorter procedures in the surgical field and improved hand–eye coordination. Visualization of interesting anatomical areas in medical images is a vital need for the IGS system which is a replacement for conventional surgery. Breast cancer is the most common cancer disease that is a leading cause of cancer-related deaths in the women population worldwide. Surgery is the initial treatment of primary breast tumors. This proposed method of non-rigid registration between pre-operative actual ultrasound and intra-operative deformed ultrasound images can minimize re-excision rates in breast-conserving surgery which is helpful to the multi-disciplinary care of breast cancer victims.

5 Results and Discussion

5.1 Dataset Description

The proposed method is applied and tested on medical image database [18] for ultrasound images of breast abnormalities [19] that was acquired by a Philips iU22 ultrasound machine at Thammasat University Hospital.

5.2 Evaluation Metrics

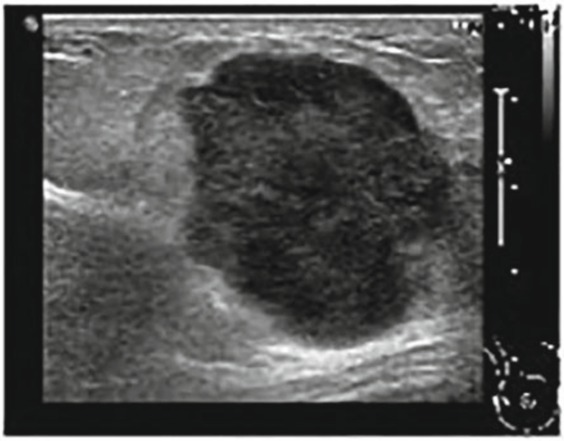
$$MSE = \frac{1}{n}\sum_{i=0}^{n}[y(x_i) - y'(x_i)]^2 \tag{15}$$

where, n: number of data, $y(x_i)$: actual values, and $y'(x_i)$: predicted values

5.3 Experimental Results

The reference image (Fig. 4) shows the actual breast tumor ultrasound taken before the operation and the test image (Fig. 5) shows the deformed breast tumor ultrasound taken during the operation. The scan conversion using the Lee filter is applied on both the ultrasound images to de-noising speckles before registration. The Speckle Index (SI) values of the original reference image and test image are 3.5e−6 and 3.6e−6, respectively. After noise filtering using the Lee filter, the SI values become 3.4e−6 and 3.5e−6 and the MSE become 29.4 and 30.5 respectively for the reference image and test image. This test image is registered according to the reference image. After the registration using the proposed methods—SSD, cubic B-spline, and L-BFGS—the MSE becomes 2.2 between the reference image and the registered test image as displayed in Fig. 6. But MSE becomes 98.5, after registration using the methods SSD, affine, and L-BFGS as shown in Fig. 7; Tables 1 and 2.

Fig. 4 Pre-operative actual breast tumor ultrasound after de-noising as reference image



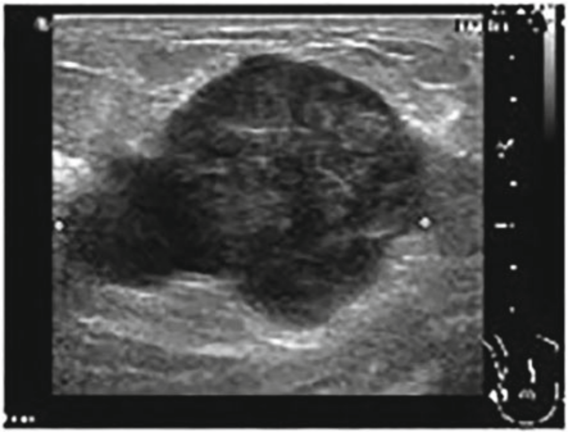


Fig. 5 Intra-operative deformed breast tumor ultrasound after de-noising as test image

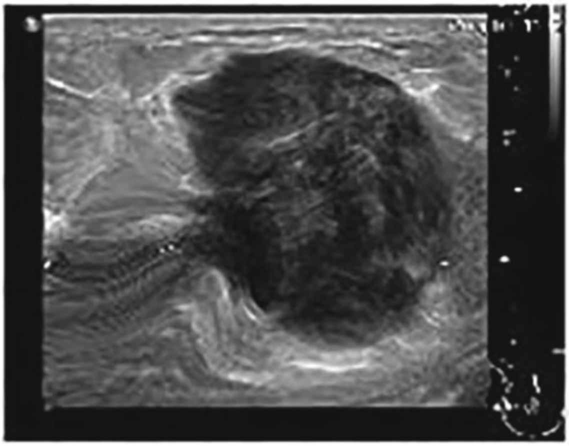


Fig. 6 Proposed Cubic B-spline method output (MSE = 2.2): Intra-operative registered test image using reference image

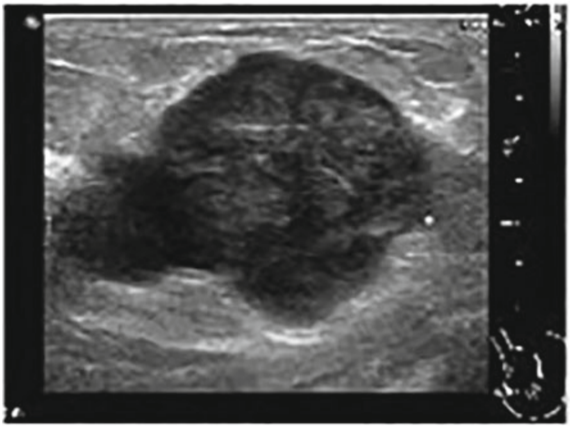


Fig. 7 Affine transformation method output (MSE = 98.5): Intra-operative registered test image using reference image

Table 1 Result of noise filtering using Lee filter on breast ultrasound images

	SI of original image	SI after noise filtering
Reference image	3.5e−6	3.4e−6
Test image	3.6e−6	3.5e−6

Table 2 Result of ultrasound non-rigid registration using cubic B-spline and Affine

Non-rigid registration between reference image and test image after image de-noising	MSE
Registered test image using cubic B-spline	2.2
Registered test image using affine transformation	98.5

6 Conclusion

This proposed method for mono-modal non-rigid registration is applied in breast-conserving surgery of breast tumors between pre-operative actual and intra-operative deformed ultrasound that can reduce re-excision rates, least fault, great success rate, and minimum risk of tissue damage which is helpful to the multi-disciplinary care of breast cancer victims. The paper also analytically derives the proposed method and optimizes it. The experimental result shows that the proposed method gives (MSE 2.2) comparable results with state-of-the-art and establishes the cubic B-spline free form deformation model which is the best in such cases while affine transformation fails in ultrasound registration.

References

1. Che C, Mathai TS, Galeotti J (2017) Ultrasound registration: a review. Methods 115:128–143
2. Wildeboer RR, van Sloun RJG, Postema AW (2018) Accurate validation of ultrasound imaging of prostate cancer: a review of challenges in registration of imaging and histopathology. J Ultrasound
3. Chel H, Nandi D, Bora PK (2015) Image registration in presence of multiplicative noise by particle swarm optimization. In: Third international conference on image information processing
4. Sotiras A, Davatzikos C, Paragios N (2013) Deformable medical image registration: a survey. IEEE Trans Med Imag 32(7):1153–1190
5. Wein W, Ladikos A, Fuerst B, Shah A, Sharma K, Navab N (2013) Global registration of ultrasound to mri using the lc2 metric for enabling neurosurgical guidance. In: Medical image computing and computer-assisted intervention (MICCAI), Springer, pp 34–41
6. Zhou H, Rivaz H (2016) Registration of pre-and postresection ultrasound volumes with noncorresponding regions in neurosurgery. IEEE J Biomed Health Inform 20(5)
7. Rivaz H, Chen SJ, Collins DL (2015) Automatic deformable MR-ultrasound registration for image-guided neurosurgery. IEEE Trans Med Imag
8. Rivaz H, Collins DL (2015) Near real-time robust non-rigid registration of volumetric ultrasound images for neurosurgery. Ultrasound Med Biol 41(2):574–587
9. Green CA, Goodsitt MM, Roubidoux MA (2020) Deformable mapping using biomechanical models to relate corresponding lesions in digital breast tomosynthesis and automated breast ultrasound images. Med Image Anal 60

10. Guo Y, Suri J, Sivaramakrishna R (2006) Image registration for breast imaging: a review. In: 27th annual IEEE conference on engineering in medicine and biology
11. Green CA, Goodsitt MM, Roubidoux MA (2018) Deformable mapping technique to correlate lesions in digital breast tomosynthesis and automated breast ultrasound images. Med Phys 45(10)
12. Saadai P, Moezzi M, Menes T (2011) Preoperative and intraoperative predictors of positive margins after breast-conserving surgery: a retrospective review. Breast Cancer 18:221–225
13. Goshtasby AA (2012) Image registration: principles, tools and methods. Springer
14. Ghosh D, Nandi D (2018) A novel speckle reducing scan conversion in ultrasound imaging system. LNCS
15. Du X, Dang J, Wang Y, Wang S, Lei T (2016) A parallel nonrigid registration algorithm based on B-spline for medical images. Comput Math Methods Med
16. Gálvez A, Iglesias A, Avila A, Otero C, Arias R (2015) Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting. Appl Soft Comput 26:90–106
17. Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. Math Program
18. Medical Image Database on Breast Ultrasound Webpage. http://www.onlinemedicalimages.com/index.php/en/site-map
19. Rodtook A, Kirimasthong K, Lohitvisate W (2018) Automatic initialization of active contours and level set method in ultrasound images of breast abnormalities. Pattern Recogn 79:172–182

Effect of IEEE 802.15.4 MAC Layer on Energy Consumption for Routing Protocol for Low Power Lossy Networks (RPL)

Aparna Telgote and Sudhakar Mande

Abstract Internet of Things (IoT) is the emerging technology responsible for the Industry 4.0 revolution, for Low Power Lossy network. The overall network lifetime depends on the power utilization of each node. This paper represents the analysis of the Radio Duty Cycle Protocol for IoT networks. The duty cycle also affects the average power utilization and node longevity, because nodes save the most energy when they are in sleeping mode, the lower the duty cycle, the longer the node's lifetime. So here are the existing RDC protocols like X-MAC, CX-MAC, ContikiMAC, Null-MAC are compared for the different RDC channel rates 8,16 and 32 Hz. And it is found that ContikiMAC reduces the duty cycle as compared to X-MAC, CX-MAC, and NullRDC as follows, from 22.52% to 6.54% for X-MAC, 27.34% to 6.54% for CX-MAC and 99% to 6.54% for NullRDC for RDC channel rate 32 MHz. As the duty cycle affects the average power utilization of the node and ultimately the overall network lifetime, the right choice of RDC depends on a protocol that is very important so ContikiMAC can be applied for real-time applications of IoT.

Keywords RDC · X-MAC · CX-MAC · ContikiMAC

1 Introduction

Small embedded devices with limited power, processing resources and memory, are rapidly using the Internet Protocol Suite to send the data to the internet, resulting in constrained-node networks, also known as Low Power Lossy Networks (LLN). The border router is regarded as an association point between the nodes/sensors and the internet. The Low Power Lossy Network (LLN) consists of numerous constraint nodes and the border router. Because of the low processing power and computing

A. Telgote
Ramrao Adhik Institute of Technology, Navi Mumbai, Mumbai, India
e-mail: aparna@dbit.in

S. Mande (✉)
Don Bosco Institute of Technology, Mumbai, India
e-mail: ssmande@dbit.in

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*, Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_16

aptitude of the constraint devices, data transmission to the internet is challenging. Furthermore, the LLN increases the difficulty of numerous tasks such as routing, network discovery, and addressing, as well as sensing and overcoming heterogeneity [1–3]. The Internet Engineering Task Force (IETE) has proposed a solution in the form of a layered architecture called 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) layer architecture to address these concerns [4]. In this paper, we have concentrated on the IEEE 802.15.4 MAC layer which consists of the Radio Duty Cycle as a sub-layer. The duty cycle has an impact on the average power consumption of node and node lifetime since nodes save the most energy when they are in the sleeping mode; the minimum the duty cycle, the longer the node's life and ultimately network's life. Along with the MAC layer, routing is a critical aspect that affects information exchange connection and performance. The selection of routing protocol and its implementation is important to calculate the overall performance of a Low Power and Lossy Network (LLN). Also, the success of the routing protocol for LLNs depends on proper utilization of limited resources, proper control of traffic, convergence Time, less Energy Consumption, low latency, and packet delivery ratio (PDR) which are all important factors in the routing protocol's success. To improve the average lifetime of the node the proper utilization of resources is very important and its found that much of the energy is wasted in transmitting and receiving data as the node needs to be in the wake-up mode to receive and send the data [5, 6]. This paper represents the analysis of the RDC protocol and MAC protocol for RPL application because the greatest power savings are gained when nodes are sleeping, and the duty cycle has an impact on average power consumption and node lifetime. The lower the duty cycle, longer the lifetime of the network.

The paper represents the research work as follows, Sect. 2. gives the background of the MAC layer. In Sect. 3 literature review is discussed. Methodology and experiment setup is discussed in Sect. 4. Results and discussion are represented in Sect. 5.

2 Background

2.1 Protocol Stack for LLN

Figure 1 shows the 6LoWPAN protocol stack which consists of 5 layers. Layer 1 is the physical layer(IEEE802.15.4), Layer 2 is the MAC(IEEE 802.16.4 MAC) and adaptation layer with 6LoWPAN protocol, Layer 3 is the network layer with RPL protocol, Layer 4 is the transport layer which has TCP and UDP protocol, Layer 5 is the application layer with CoAP protocol. This research is based on the MAC layer and protocol related to it

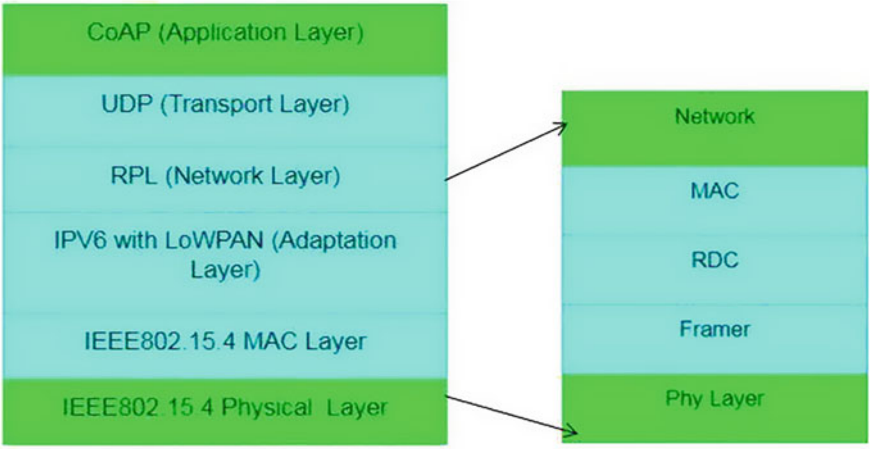


Fig. 1 To overcome the problems of constrained devices of the Low Power Lossy network IETF has suggested the protocol architecture which has different protocols as compared to the traditional TCP/IP network and also between the physical layer and the network layer 3 more protocols exist called as Framer, Medium Access Control, Radio Duty Cycle

2.2 IEEE 802.15.4 Medium Access Control (MAC)

The medium access control (MAC) layer is responsible for fairness, high throughput with minimum energy consumption, and low latency. The MAC layer is also specified in IEEE 802.15.4 standard. Long (64-bit) and short (32-bit) MAC addressing are supported by IEEE 802.15.4. (16 bit). Quadrature phase-shift keying (QPSK) and binary phase-shift keying (BPSK) are the types of modulation used by IEEE 802.15.4. It has a maximum 128-byte frame size, including the (1 byte) MAC header [2, 4, 7].

Framer, Medium Access Control and Radio duty cycle are the sub layers of IEEE 802.15.4 MAC layer. The variables such as NETSTACK framer, NETSTACK RDC, NETSTACK MAC are used by the network layer respectively. The framer layer is a set of auxiliary's function used to form the frame which contains data to be sent and received. Radio Duty Cycle (RDC): This layer controls the node's sleep period. The function of RDC is to check when to transmit the packets to also ensure that the node should be awake to receive the packet. Finally, the MAC layer is responsible for packet addressing and retransmission [8].

2.3 RDC Protocol

The following are the types of RDC protocols mentioned in the literature: X-MAC, CX-MAC, ContikiMAC, NullMac, LPP X-MAC As per paper [9], X-MAC is TinyOS' (WSN) well accepted RDC protocol. In this protocol whenever the sender wants to send a frame, before sending the actual data it sends a small request frame to its destination and waits for the ACK from the destination. After receiving this small frame, the receiver sends the ACK, After receiving ACK, the sender sends the Data [4, 5, 10]. Effect of IEEE 802.15.4 MAC layer on energy consumption for routing protocol for low power lossy networks (RPL).

CX-MAC. It's called Contiki X-MAC. It works on the principle of phase lock-in where when the sender and receiver are in phase it means that the receiver is in the listing mode and the sender transmits the probe.

ContikiMAC. The receiver sleeps most of the time, but it wakes up a few times every second to execute two Clear Channel Assessments (CCA). In between two CCA pairs, there is a "Wake Up Interval" (WUI) of 125 ms. If no radioactivity is detected during two consecutive CCAs, the receiver can go back to the sleep mode. Whenever CCA detects the radioactivity, the receiver remains awake to receive the frame and sends ACK the frame received, and goes back to sleep. Before going back to the sleep mode, thereceiver will wait a long time for successive frames to be delivered in the same time slot and send ACK. The number of retransmissions should be kept to a minimum to minimize power utilization at the sender. When a sender receives ACK, it will come to know that the receiver was sleepless, right before sending the acknowledged frame. Because the wake-up periods are periodic with an interval of WUI, the sender will create a learning table that specifies the best time for each destination to begin the transmission of a frame (by locking the phase of the transmitter and receiver). Once the locking has been set up, rather than retransmitting the frame, again and again, it will only retransmit a few times because retransmission begins only when the receiver is about to awake. The phase lock can work for a long time though tx and Rx clocks are not synchronized by updating the timing when getting each ACK [10].

NullRDC: Contiki provides NullRDC, a radio duty cycling protocol configured like the other RDC protocols but this protocol keeps the radio ON all the time.

3 Medium Access Control Protocol for Contiki OS

Contiki supports CSMA, NullMAC as MAC protocols.

3.1 Carrier Sense Multiple Access (CSMA)

Because the two more efficient versions of the Aloha-derived medium access control protocols (CSMA/CD and CSMA/CA) do not meet the need for constraint devices because of less memory, and processing power and therefore according to [11] the author described that "Contiki has implemented a simplified CSMA protocol that manages a separate FIFO queue for each possible destination and tries to transmit each frame multiple times before dropping the frame after three unsuccessful attempts". In CSMA, the time between two successive transmission attempts should be kept to a minimum. The time between subsequent transmission attempts in CSMA should be random, with a mean that grows exponentially. "The exponential function is simply approximated by a linear function" because the maximum number of trials is limited [7].

3.2 Application and Router Layers

Above the MAC and RDC layers, Contiki provides two different groupings of protocols. The first is known as Rime. It is a Contiki-specific communication protocol, while the second, known as uIP, is an adaptation of the IP layer architecture for constraint nodes [10].

4 Literature Review

The power management of hardware platforms is one of the most difficult difficulties in deploying IoT applications. Researchers are now employing a variety of novel technologies to reduce radio power consumption and to open the door of IoT for real-time applications. The hardest job in IoT systems is to find the power utilization of Tx and Rx when they are communicating with each other [4]. Because in the nodes' communication, when the radio is in a sleepless state, a lot of effort has gone into developing "power-efficient radio WakeUp models". Various techniques based on hardware and software methods to be used to manage the radio WakeUp mode have been suggested [12–14]. The authors in [15] designed a new IoT node for "long and short-range networking" using a mix of energy scavenging WakeUp receiver and "LoRa radio technology". "BLE technology and WakeUp radio" are combined with energy scavenging in another approach. Both proposed solutions are hardware-based, with a "dual-radio mechanism" utilizing different components in the Node radio structure to form SoC (System on Chip)", which is costly for IoT devices. Small hardware size and low cost are critical variables for adopting WSN on a big scale. Other protocol-based approaches, also [12, 14], have been introduced which are more reactive to change of the channels [12]. As per literature [16] "The average power utilization of each device (P) is the sum of the average power utilization in the CPU state (Pcpu)- is activated whenever the device is active (the CPU is active without using the radio transceiver is CPU Tx – Rx)", the Low Power Mode state (Pump)- is ON when the sensor device goes into low power state, the Rx state (PRx)- the sensor device is ON in the radio receive state, and based on the platform data sheet, the voltage level (VCC) and current power utilization in the stated condition are established. The power utilization in each state is determined using the number of “CPU ticks” depending on the use of the “microcontroller”, the “current power consumption” in the stated state, and the “battery” [17, 18].

In this paper, we have analyzed all the RDC and MAC protocols for 30 node topologies which consist of 30 UDP clients and 1 UDP server. The application is the Routing Protocol for Low Power Lossy Network(RPL) in RPL as nodes want to send the data to the root node or sink node to form a DODAG. During this process lots of control messages are exchanged between the nodes to find the best parents as per the objective function. And because of this overhead of messages, the power consumption of the node will increase, and ultimately the life of the network will decrease. So it is important to choose the correct MAC protocol to improve the overall life of the network. This paper gives a comparison of all the existing MAC protocols.

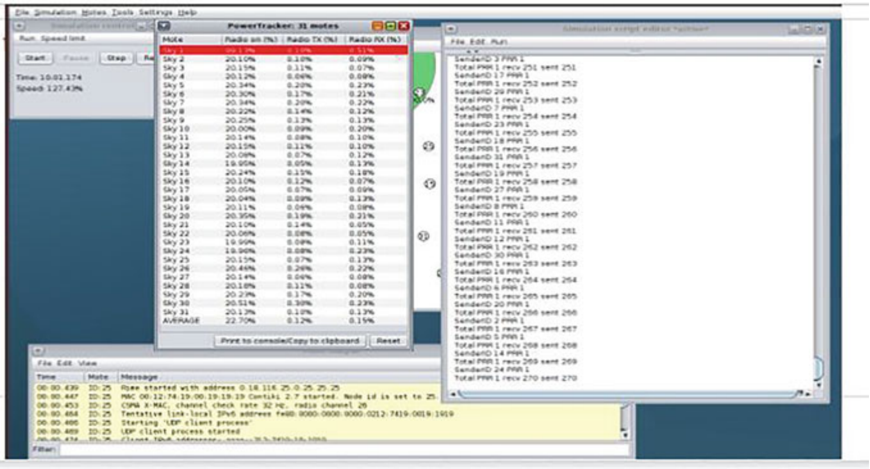
5 Methodology

5.1 Contiki OS

The latest version of Contiki 2.7 is used in this investigation. Contiki, as previously said, is an open-source Internet of Things operating system that enables tiny low-cost, low-power embedded devices connected to the Internet. We concentrate on Contiki’s offered features, to give an optimum technique to reduce energy usage, and network protocols and radio duty cycles were combined. The authors in [11] explain about the Cooja simulator software, in IoT nodes where the Cooja simulator, which is available at Contiki OS westie, is used in this study to model the topology and run the proposed approach on the nodes in the network scenario. During compilation, all three layers (“Framer, RDC, and MAC”) must be declared. They are defined in the Makefile or the project-conf.h file. File “core/net/netstack.h” contains the specifications of protocols used. It binds the “NETSTACK FRAMER, NETSTACK RDC, and NETSTACK MAC” to a protocol that can be used further [11, 19] (Fig. 2 and Table 1).

6 Result and Discussion

The topology of 30 nodes and 1 sink was created using the Contiki OS cooja simulator, and the power trace code was written to trace the power of each node.


File Simulation Motes Tools Settings Help
Run Speed limit
Start
Pause
Step
File Edit Run
File Edit View
Print to console/Copy to clipboard
Reset
Filter:

Time: 10:01.174
Speed: 127.43%

PowerTracker: 31 motes

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	20.13%	0.10%	0.51%
Sky 2	20.10%	0.10%	0.09%
Sky 3	20.15%	0.11%	0.07%
Sky 4	20.12%	0.06%	0.08%
Sky 5	20.34%	0.20%	0.23%
Sky 6	20.30%	0.17%	0.21%
Sky 7	20.34%	0.20%	0.22%
Sky 8	20.22%	0.14%	0.12%
Sky 9	20.25%	0.13%	0.13%
Sky 10	20.00%	0.09%	0.20%
Sky 11	20.14%	0.08%	0.10%
Sky 12	20.15%	0.11%	0.10%
Sky 13	20.08%	0.07%	0.12%
Sky 14	19.95%	0.05%	0.13%
Sky 15	20.24%	0.15%	0.18%
Sky 16	20.10%	0.12%	0.07%
Sky 17	20.05%	0.07%	0.09%
Sky 18	20.04%	0.09%	0.13%
Sky 19	20.11%	0.06%	0.08%
Sky 20	20.35%	0.19%	0.21%
Sky 21	20.10%	0.14%	0.05%
Sky 22	20.06%	0.08%	0.05%
Sky 23	19.99%	0.08%	0.11%
Sky 24	19.96%	0.08%	0.23%
Sky 25	20.15%	0.07%	0.13%
Sky 26	20.46%	0.26%	0.22%
Sky 27	20.14%	0.06%	0.08%
Sky 28	20.18%	0.11%	0.08%
Sky 29	20.23%	0.17%	0.20%
Sky 30	20.51%	0.30%	0.23%
Sky 31	20.13%	0.10%	0.13%
AVERAGE	22.70%	0.12%	0.15%

```
SenderID 3 PRR 1
Total PRR 1 recv 251 sent 251
SenderID 17 PRR 1
Total PRR 1 recv 252 sent 252
SenderID 29 PRR 1
Total PRR 1 recv 253 sent 253
SenderID 7 PRR 1
Total PRR 1 recv 254 sent 254
SenderID 23 PRR 1
Total PRR 1 recv 255 sent 255
SenderID 18 PRR 1
Total PRR 1 recv 256 sent 256
SenderID 31 PRR 1
Total PRR 1 recv 257 sent 257
SenderID 19 PRR 1
Total PRR 1 recv 258 sent 258
SenderID 27 PRR 1
Total PRR 1 recv 259 sent 259
SenderID 8 PRR 1
Total PRR 1 recv 260 sent 260
SenderID 11 PRR 1
Total PRR 1 recv 261 sent 261
SenderID 12 PRR 1
Total PRR 1 recv 262 sent 262
SenderID 30 PRR 1
Total PRR 1 recv 263 sent 263
SenderID 16 PRR 1
Total PRR 1 recv 264 sent 264
SenderID 6 PRR 1
Total PRR 1 recv 265 sent 265
SenderID 20 PRR 1
Total PRR 1 recv 266 sent 266
SenderID 2 PRR 1
Total PRR 1 recv 267 sent 267
SenderID 5 PRR 1
Total PRR 1 recv 268 sent 268
SenderID 14 PRR 1
Total PRR 1 recv 269 sent 269
SenderID 24 PRR 1
Total PRR 1 recv 270 sent 270
```

```
Time       Mote   Message
00:00.439  ID:25  Rime started with address 0.18.116.25.0.25.25.25
00:00.447  ID:25  MAC 00:12:74:19:00:19:19:19 Contiki 2.7 started. Node id is set to 25.
00:00.453  ID:25  CSMA X-MAC, channel check rate 32 Hz, radio channel 26
00:00.464  ID:25  Tentative link-local IPv6 address fe80:0000:0000:0000:0212:7419:0019:1919
00:00.466  ID:25  Starting 'UDP client process'
00:00.469  ID:25  UDP client process started
```

Fig. 2 Simulation setup: The Cooja simulator in the configuration consists of 30 nodes as a UDP client node and 1 UDP server node, where all the clients wanted to reach to the root node or server node to form DODAG. The top left panel shows simulation control that allows us to see the time lapse, the speed of simulation and start, pause, step and reload button. The bottom window shows mote output that network layer used Rime, MAC layer used CSMA protocol, RDC layer used X-MAC protocol with the channel rate of 32 Hz and the message was sent by the UDP client. The window below the mote output is the power tracker that allows to track the motes radio power with a seperate transmitter and receiver

Table 1 Evaluation parameters

Parameters	Value
Application	Powertrace
Transport Layer Protocol	UDP
Network stack	rime
MAC protocol	CSMA, NullMAC
Radio Duty Cycle (RDC) Protocol	ContkiMAC, X MAC, CXMAC, NullRDC
Sleep Cycle	8 Hz, 16 Hz, 32 Hz
No. of nodes	31
Area	100 m × 100 m
Physical Layer	802.15.4
MAC layer	IEEE 802.15.4 (ContikiMAC)
Network Layer (Routing Protocol)	RPL

6.1 How to Read the Output of Powertrace File

The explanation of the output of the powertrace file is given as follows, 60,445 P 0.18 16 118,490 1,847,789 15,413 130,581 135 110,321 118,490 1,847,789 15,413. 130,581 135 110,321 (radio 7.42%/7.42%tx 0.78%/ 0.78%listen 6.64% / 6.64%) Contiki estimates the nodes' power usage using the Energest software-based module. This module measures the amount of time spent by each sensor node in real-time in phases such as CPU, LPM, Tx, and Rx. When the node is powered on, the "energy estimation module" is invoked to generate a "time stamp". The Power utilization of the node can be calculated by

$$P = Pcpu + Plmp + PRx + PTx.$$

Where P = Total power utilization of node.

Pcpu = Power utilization of CPU.

Plmp = Power utilization when the device is in low power state.

PRx = Power utilization when the device is in listen state.

PTx = Power utilization when the device is in transmit state (Table 2).

Table 2 How to read the output of powertrace file

Parameters of powertrace file	Explanation of each parameter
clock time = 60,445	Clock time
rimeaddr = 0,18	Rime address
seq. no. = 16	Sequence number
all cpu = 118,490	Accumulated CPU power utilization
all lpm = 1,847,789	Accumulated low power mode power utilization
all transmit = 15,413	Accumulated transmission's power utilization
all listen = 130,581	Accumulated listen power utilization
all idle transmit 135	Accumulated idle transmission power utilization
all idle listen = 110,321	Accumulated idle listen power utilization
cpu = 118,490	CPU power utilization for this cycle
lpm = 1,847,789	LPM power utilization for this cycle
transmit = 15,413	Transmission power utilization for this cycle

Table 3 TmoteSky parameters and its power utilization as per data sheet

Parameters	Power utilization state	Voltage requirement
VCC	"Supply voltage"	3 V
P CPU	"MCU on, Radio off"	1.8 mW
P LPM	"MCU idle, Radio off"	0.0545 mW
P Tx	MCU on, Radio Tx	17.7 mW
P Rx	MCU on, Radio Rx	20 mW

According to the data sheet for TMoteSky the following parameters need to be considered (Table 3).

Number of ticks per second for rtime (RTIMER SECOND = 32,768)

The Voltage requirement for TmoteSky is 3 V.

As per [20] the following formula is used to calculate the power consumption for each mote after getting the rtime ticks in each state is

> "Energy (mJ) = (Transmit * 19.5 mA + Listen * 21.5 mA + CPU time * 1.8 mA + LPM * 0.0545 mA) * 3 V / (32,768)"

6.2 Rtimer Arch Per Second

In Figs. 3, 4, 5, the Y axis indicates Rtimer Arch per second indicating the number of ticks utilized by the nodes when they are transmitting data, receiving data, in ideal mode or in Low power mode for channel check rate 8, 16, 32 Hz.

6.3 Energy Consumption for Channel Check Rate 8 Hz

From this Fig. 6, it is observed that if we use MAC protocol as CSMA and RDC protocol as X-MAC, CX-MAC, ContikiMAC, for channel check rate 8 Hz, power utilization of X-MAC is 149% more, CX-MAC is 218% more and power utilization of NullRDC is 2800% more as compared to ContikiMAC.

6.4 Energy Consumption for Channel Check Rate 16 Hz

From this Fig. 6, it is observed that if we use MAC protocol as CSMA and RDC protocol as X-MAC, CX-MAC, ContikiMAC, for channel check rate 16 Hz, power utilization of X-MAC is 286% more, CX-MAC is 391% more and power utilization of NullRDC is 2657% more as compared to ContikiMAC.

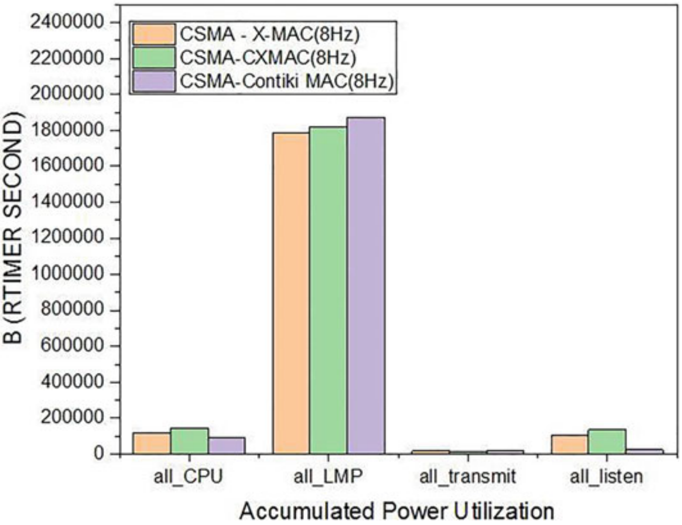


Fig. 3 Accumulated energy Consumption of CPU, LMP, TX, RX, Idle Tx and Idle listen for channel check rate 8 Hz. Y axis indicates Rtimer Arch/sec., means no. ticks /sec, from this figure. Its clear that more no. of ticks are utilized when the device is in the low power mode and therefore there will be more energy consumption ad also ContikiMAC power utilization will be less in all CPU mode, all transmit mode, and all Received mode as compared to X-MAC and CX-MAC but more in all LMPs CX-MAC utilized less power while CX-MAC needed more power in all CPU mode and all listen mode

6.5 Energy Consumption for Channel Check Rate 32 Hz

From this Fig. 6, it is observed that if we use MAC protocol as CSMA and RDC protocol as X-MAC, CX-MAC, ContikiMAC, for channel check rate 32 Hz, power utilization of X-MAC is 417.4% more, CX-MAC is 554.7% more and power utilization of NullRDC is 2057% more as compared to ContikiMAC.

7 Conclusion

The Internet of Things (IoT) is one of the emerging technologies responsible for the Industry 4.0 revolution. For Low Power Lossy network the overall network lifetime depends on the power consumption of each node. This report represents the analysis of the Radio Duty Cycle Protocol for IoT networks. The duty cycle also affects the average power consumption and node longevity, because nodes save the most energy when they are sleeping lower; the lower the duty cycle, the longer the node's lifetime. So here are the existing RDC protocols such as X-MAC, CXMAC, ContikMAC, and NullRDC compared for RDC channel rates 8, 16, and 32 Hz. And it is observed that if we use the MAC protocol as CSMA and RDC protocol as X-MAC, CX-MAC,

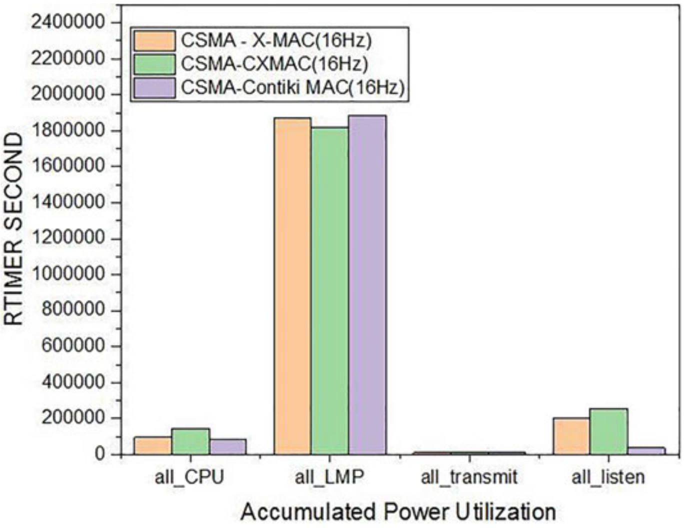


Fig. 4 Accumulated energy Consumption of CPU, LMP, TX, RX, Idle Tx and Idle listen for channel check rate 16 Hz. From this figure it is clear that more no. of ticks are utilized when the device is in low power mode and therefore there is more energy consumption in the LMP mode. Another observation is that the ContikiMAC power utilization is less for all CPU mode, all transmit mode, all Received mode as compared to X-MAC and CX-MAC but in all LMP power utilization of X-MAC and ContikiMAC is the same, while CX-MAC utilized less power. CX-MAC needs more power in all CPU mode and all listen mode

ContikiMAC and NullMAC, for channel check rate 8 Hz, power utilization of X-MAC is 149% more, CX-MAC is 218% more and power utilization of NullRDC is 2800% more as compared to ContikiMAC. Similarly for channel check rate 16 Hz, power utilization of X-MAC is 286% more, CX-MAC is 391% more and power utilization of NullRDC is 2657% more as compared to ContikiMAC. And for channel check rate 32 Hz, power utilization of X-MAC is 417.4% more, CX-MAC is 554.7% more and power utilization of NullRDC is 2057% more as compared to ContikiMAC. So if we use CSMA as the MAC protocol and ContikiMAC as the RDC protocol lots of power of the individual node can be saved. This can increase the life of network, so can be considered for real time IoT applications.

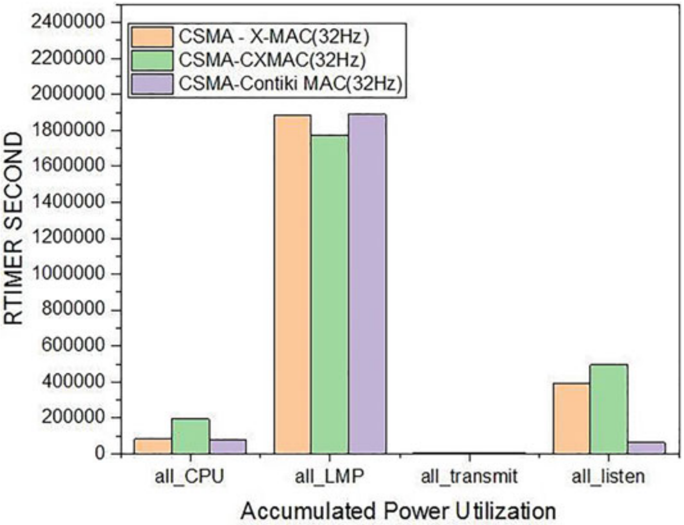


Fig. 5 Accumulated energy Consumption of CPU, LMP, TX, RX, Idle Tx and Idle listen for channel check rate at 32 Hz. From this figure it is clear that more no. of ticks are utilized when the device is in low power mode and therefore more energy consumption in the LMP mode another observation is that the ContikiMAC power utilization is less for all CPU mode, all transmit mode, all Received mode as compared to X-MAC and CX-MAC but in all LMP power utilization of X-MAC and ContikiMAC is the same, while CX-MAC utilized less power. CX-MAC needs more power in all CPU mode and all listen mode

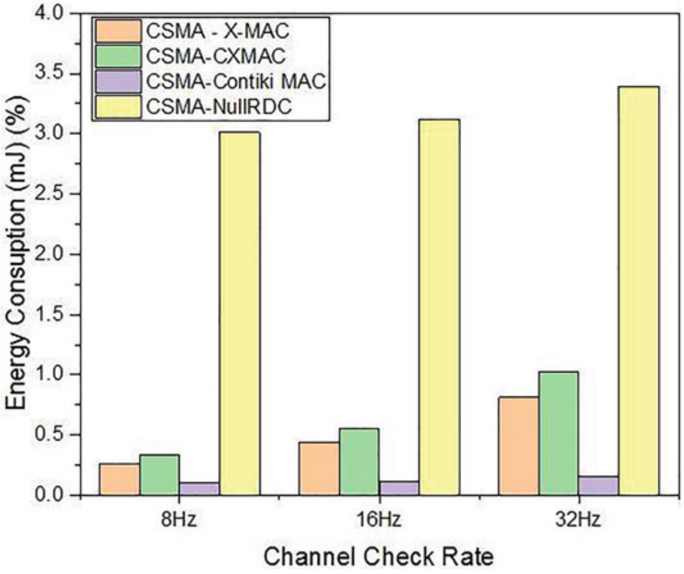


Fig. 6 Energy Consumption for channel check rate 8, 16 and 32 Hz

References

1. Demirkol CE, Alagoz F (2006) MAC protocols for wireless sensor networks: a survey. IEEE Commun Mag 44(4):115–121
2. Kabara J, Calle M (2012) MAC protocols used by wireless sensor networks and a general method of performance evaluation. Int J Distrib Sens Netw
3. Haseeb S, Hashim AH, Khalifa OO, Ismail AF (2017) Connectivity, interoperability and manageability challenges in internet of things. In: AIP conference proceedings, vol 1883, p 020004. https://doi.org/10.1063/1.5002022
4. Omer Farooq M (2020) RIoT: a routing protocol for the internet of things. Comput J 63(1): 958–973. https://doi.org/10.1093/comjnl/bxaa012
5. Dunkels A (2011) The ContikiMAC radio duty cycling protocol. Technical report, Swedish Institute of Computer Science (SICS), 2011
6. Buettner M, Yee GV, Anderson E, Han R (2006) X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, department of computer science University of Colorado, USA, 2006
7. Farooq MO, Kunz T (2015) Contiki-based IEEE 802.15.4 channel capacity estimation and suitability of its CSMA-CA MAC layer protocol for real-time multimedia applications. Mob Inf Syst
8. Safaei B, Monazzah AMH, Ejlali A (2021) ELITE: an elaborated cross-layer RPL objective function to achieve energy efficiency in internet-of-things devices. IEEE Internet Things J 8(2):1169–1182. https://doi.org/10.1109/JIOT.2020.3011968
9. Magno M, Aoudia FA, Gautier M, Berder O, Benini L (2017) WULoRa: an energy efficient IoT end-node for energy harvesting and heterogeneous communication. In: Proceedings of 2017 design, automation, and test in Europe DATE 2017, 2017, pp 1528–1533
10. Maite Martincorena Arraiza (2015) RDC protocols in wireless sensor networks running Contiki. Thesis
11. https://anrg.usc.edu/contiki/index.php/MAC, protocolsin ContikiOS
12. Guo C, Zhong LC, Rabaey JM (2001) Low power distributed MAC for ad hoc sensor radio networks. In: Proceedings of the GLOBECOM'01. IEEE global telecommunications conference (Cat. No. 01CH37270), Oslo, Norway, 25–29 November 2001
13. Mahlknecht S, Durante MS (2009) WUR-MAC: anergy efficient wakeup receiver based MAC protocol. IFAC Proc 42:79–83
14. Amirinasab Nasab M, Shamshirband S, Chronopoulos AT, Mosavi A, Nabipour N (2020) Energy-efficient method for wireless sensor networks low-power radio operation in internet of things published in MPDI Journal 2020.
15. Basagni S (2016) CTP-WUR: the collection tree protocol in wake-up radio WSNs for critical applications. In Proceedings of the 2016 international conference on computing, networking and communications (ICNC), Beijing, China, 4–6 June 2016
16. Amirinasab Nasab M, Shamshirband S, Chronopoulos AT, Mosavi A, Nabipour N (2020) Energy-efficient method for wireless sensor networks low-power radio operation in internet of things. published in MPDI J
17. Berkeley's OpenWSN Project. http://openwsn.berkeley.edu/
18. Nano-RK. Available online. http://www.nanork.org/projects/nanork
19. Cotiki: The Open Source Operating System for the Internet of Things. http://www.contiki-os.org/
20. Eloudrhiri Hassani A, Sahel A, Badri A (2019) Impact of RPL objective functions on energy consumption in Ipv6 based wireless sensor networks. Institut Universitaire de Technologied'Aix-Marseille (France), Jun 2019, CASABLANCA, Morocco
21. Internet Engineering Task Force (IETF) Request for Comments: 6550: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. ISSN: 2070-1721, 2012
22. Ahmed Z, Rehan M, Chughtai O, Rehan W AD-RDC: a novel adaptive dynamic radio duty cycle mechanism for low-power IoT devices. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2022.3145017.

23. Tripathi J, Oliveira J, Vasseur J-P (2012) Performance evaluation of the routing protocol for low- power and lossy networks (RPL). IETF RFC 6687:1–26
24. Ko J, Eriksson J, Tsiftes N, Dawson-haggerty S, Terzis A, Dunkels A, Culler D (2011) ContikiRPL and TinyRPL: happy together. In: Proceedings of the workshop on extending the internet to low power and Lossy Networks (IP+SN), Chicago, IL, USA, 11 April 2011
25. Tsiftes N, Eriksson J, Dunkels A (2010) Low-power wireless IPv6 routing with Con- tikiRPL. In: Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks, IPSN '10, Stockholm, Sweden, 12–16 April 2010, ACM Press: New York, NY, USA, 2010; p 406
26. Jeong J (2011) Design and implementation of low power wireless IPv6 routing for NanoQplus. In: Proceedings of the 13th international conference on advanced communication technology (ICACT), Daejeon, South Korea, 13–16 February 2011, pp 966–971
27. Amirinasab Nasab M, Shamshirband S, Chronopoulos AT, Mosavi A, Nabipour N (2020) Energy-efficient method for wireless sensor networks low-power radio operation in internet of things. Electronics 9. https://doi.org/10.3390/electronics9020320

mCNN: An Approach for Plant Disease Detection Using Modified Convolutional Neural Network

S. Brinthakumari and P. M. Sivaraja

Abstract Plant disease is a persistent problem for farmers, and it is one of the most serious risks to income and food security. This initiative aims to cultivate the productivity of agricultural output in the nation by classifying plant leaves into sick and healthy leaf types. The smart farming system is an innovative technology that aids in the improvement of agricultural quality and quantity. Deep learning using Convolutional Neural Networks (CNN) has successfully classified various plant leaf diseases. It represents a contemporary technique that offers cost-effective disease diagnosis. CNN presents a simplified version of a much broader image. In this paper, we proposed a hybrid novel model for detection of plant diseases using mixed Deep Learning (DL) procedures. The UNET based DL framework was used for disease detection and classification. In convolutional neural layer feature extraction was done and the pooling layer optimized those features, and finally a dense layer classifies the test object. Numerous synthetic and real time plan dataset have been used for evaluation. In extensive experimental analysis two ML and two DL classifiers are implemented such as SVM, PCA, CNN and modified CNN (mCNN). The mCNN is the collaboration of VGG16 and VGG16 backbone for classification and YOLOv3 model data pre-processing. The mCNN obtains 96.80% detection and classification accuracy on a heterogenous dataset which is higher than other classifiers as well as conventional classifiers.

Keywords Plant disease prediction · Deep CNN · Machine learning · Feature extraction and selection · Supervised classification

S. Brinthakumari (✉)
Department of Information Technology, K.C College of Engineering and Management Studies and Research, Thane, Maharashtra, India
e-mail: brinthakumaris@kccemsr.edu.in

P. M. Sivaraja
Department of Computer Science and Engineering, Amrita College of Engineering and Technology, Erachakulam, Tamilnadu, India

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*, Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_17

1 Introduction

Our Nation's Economy is based on agriculture. It supports over 70% of the population and the actual GDP. India is the greatest producer of pulses, rice, wheat, spices, and spice products globally. Any country's agriculture is dependent on the quality and quantity of agricultural goods, particularly plants. Numerous researchers used image processing, the ML and DL approaches to identify plant illness (i.e., aberrant growth or malfunction) to make this challenging work easier. Plant and tree health monitoring and disease detection are crucial for long-term agriculture. Plant diseases and pests pose a significant threat to agriculture. Plant diseases generate considerable productivity and economic losses in the agricultural business across the globe. One of the most critical objectives in overall crop disease management is to identify plant illnesses at an earlier stage to avert a more significant loss. Plant disease diagnosis entails a large amount of intricacy, which is accomplished by visual inspection of symptoms on plant leaves. Even skilled agronomists and plant pathologists often fail to detect particular illnesses due to this intricacy and the enormous number of farmed plants and their current issues, leading to incorrect conclusions and treatments. According to the findings, climatic change may affect pathogen growth stages and host resistance rates, resulting in physiological alterations in host–pathogen interactions. The problem is made even more difficult because illnesses are now more readily transmitted internationally. New diseases may emerge in regions where they have never been seen before and where there is, by definition, no local competence to fight them. One of the foundations of exactness breeding is the timely and precise identification of plant illnesses [1]. Solving the prolonged pathogen resistance improvement problem and avoiding the refusal repercussions of season change is vital to minimize inefficient waste of financial and other resources, resulting in well output. Plant diseases may be identified through several approaches. Some conditions have no evident signs, or the harm becomes apparent too late to intervene, demanding a comprehensive study. However, as most illnesses present themselves in the visible spectrum, a qualified professional's naked eye examination is the most prevalent way of diagnosing plant diseases in practice. A plant pathologist must have extraordinary observation abilities to notice specific indicators to diagnose plant diseases properly [2, 3].

Plant disease symptoms may be noticed in dissimilar areas of the plant. Nevertheless, leaves are the most typically encountered components for identifying illness. Consequently, researchers have labored to automate identifying and categorizing plant illnesses using leaf photographs. Artificial intelligence, machine learning, deep learning, image processing, and Graphics Processing Units (GPUs) may aid in expanding and increasing plant protection and growth. The usage of artificial neural network topologies with multiple processing layers is called deep learning. CNN is the principal deep learning technique utilized in this research. The CNNs are among the most effective algorithms for simulating intricate processes and performing pattern identification in applications involving massive data, such as image recognition. The major contribution of this research is listed below.

- To extract various features from the train or test image data set and build a robust model for better classification accuracy.
- We also develop a hybrid classification algorithm that collaborates with a CNN or RNN to detect heterogeneous plant diseases.
- We also support the acceptance of heterogeneous image data set that includes plant leaf disease and fruit disease for the validation plant dataset.

The rest of the paper describes the literature survey that has been demonstrated in Sect. 2, along with a focus on literature of review and Sect. 3 demonstrates the planned implementation strategy with a defined research methodology. Section 4 presented with algorithm details are used for the proposed implementation. Section 5 discussed the experimental setup and results achieved on various datasets with the hybrid classification model. Finally, in Sect. 6, we demonstrate the proposed research's conclusion and future work.

2 Literature Survey

Plant diseases diminish crop quality and productiveness, making them a key concern in agriculture. Plant diseases may cause mild to heavy damage to broad parts of planted crops, resulting in considerable financial losses and affecting the agricultural economy [1]. To avert huge losses, many disease-diagnosis methods have been created. Molecular biology and immunology can identify causal agents. Many farmers can't apply these tactics since they need specialized knowledge or a lot of money and resources. According to the UN's Food most farms are operated by families in poor countries. These homes feed much of the world's population. Poverty, food insecurity, and limited market and service access persist [2]. Many studies have been done to produce precise, farmer-friendly processes.

Precision agriculture uses advanced technology to enhance decision-making [3]. Modern digital technology gathers a lot of data in real-time, and machine learning algorithms are used to make cost-effective decisions. This topic needs advancement, especially in decision-support systems that convert enormous amounts of data into useful ideas. Several techniques and approaches may be used for this, numerous machine learning and deep learning techniques such as ANN, SVM, NB, RF etc. Deep learning (DL) methods have risen in agriculture. Computer vision and AI may provide new solutions. These methodologies make forecasts more accurately than traditional methods, improving decision-making. Advances in hardware technologies allow DL to handle complex problems quickly. These discoveries aren't trivial. DL is a cutting-edge method for categorizing land cover and may have additional uses. Deep neural networks (DNNs) perform well in hyperspectral analysis [4]. CNNs perform well in crop categorization [5, 6], quality analysis disease detection and classification using computer vision techniques [7, 8]. AlexNet [9] and GoogLeNet [10] showed existing methods performance in various investigations [11–15]. Pretrained networks perform better [13].

A comprehensive collection should comprise diverse images. Generative adversarial networks (GANs) [16] may produce synthetic data when the training material is insufficient. Existing DL solutions for plant disease detection are successful, but there's need for improvement. Traditional machine learning approaches are used to identify ailments [17]. The SVM classifiers to identify healthy and Bakanae-infected rice seedlings. The authors found the recommended procedure to be less subjective and time-consuming than standard naked-eye assessment. Another study [18] employed three classifiers: SVM, KNN, and probabilistic neural network to decrease human involvement. The authors emphasized feature extraction, background removal, and segmentation. 19 agreed. According to [19] plant disease detection and segmentation was used using ML and segmentation methods on a heterogeneous dataset.

Similarly, only a few studies have been observed at sophisticated training strategies; for example, authors in [13] looked at the performance of various deep learning frameworks that are trained from scratch as well as transfer learning methodologies. By differentiating the state-of the art DL structures for the categorization of crop disease; comparison research was done to demonstrate the relevance of the fine-tuning approach [20]. The most current discoveries in the field of plant disease categorization are given in detail.

The categorization and location of objects are conducted in a single platform utilizing deep learning meta-architectures to meet the issue of object identification. Few DL algorithms have been created in this area. The Region-based Convolution Neural Network (RCNN) was one of the earliest current algorithms to use CNN for image detection [21]. Following that, the effective application of regional proposal approaches demonstrated important advancements in object recognition. Few research works have been undertaken to execute this difficult agricultural activity using DL approaches in the situation of plant disease identification. Deep learning replicas were used to conduct plant disease localization and diagnosis, for example, in [22]. The authors effectively acquired a greater mean average accuracy by using their own annotated photos of tomato leaves. Two alternative ways to performing automated pest identification based on ML/DL learning algorithms were devised and compared in [23]. The goal of this study was to find the pest in greenhouse tomato and pepper plants. Their results revealed that deep learning approaches outperformed machine learning algorithms because of their capacity to conduct detection and classification tasks in a single step. The Single Shot Detector (SSD) was used in recent research to identify illness in Cassava leaves, and the findings were good [24]. CNN's plant disease identification task was used in another recent study to quantify the degree of abnormalities in plant leaves [25].

A few datasets have been created and utilized for a variety of real-world procedures with a large number of classes. For example, the ImageNet dataset [26], which contains an unparalleled amount of photos, has lately achieved achievements in object categorization and detection research. Similarly, the MS Coco dataset [27] has 91 object classes, 82 of which have over 5 k tagged examples each. In 328 k images, a sum of 2500 k data instances is flagged. When collated to the ImageNet (3.0) and Pascal (2.3) datasets, the MS coco dataset includes further object instances. As a result, for transfer learning, we employed the MS Coco dataset's training weights. Following that, the Plant Village dataset [28] was chosen since it includes photos that are relevant to the study region.

3 Proposed System Design

This system evaluates plant disease detection and classification using a modified convolutional neural network-based deep learning algorithm. According to Fig. 1, initially, we collect the data from numerous sources, such as some synthetic data sets or a few real-time data sets. The preprocessing has been done using noise removal, and misclassified instances have been removed in the normalization phase. In an actual convolutional neural network, the conventional layer works to extract features, while optimization is done in the feeling layer. The deep convolutional Framework has been used with n number of epoch sizes. Finally, according to the trained model, the dense layer classifies the entire test in instances. In the below section, we determine each phase of the our model concerning the module.

Image Data acquisition

This module collects plant image datasets from various sources, such as real-time validated image datasets and some synthetic datasets. The dataset may be imbalanced sometime, and it contains noisy images. In the second section, we pre-processed and normalized the entire dataset to achieve the best results for training and testing.

Data Pre-processing

Pre-processing reduces distortion, making post-processing simpler. Pre-processing includes color space transformation, cropping, smoothness, and enhancement. This module's use varies with image quality. Color space converting is accompanied by filtering and augmentation. If photographs are obtained in an uncontrolled setting with complicated backdrops, cropping is also necessary. It can be done manually or automatically with the help of functions.

Segmentation

Along with the items of interest, segmentation separates the image into sections with strong association. The number of histogram peaks, for example, is one feature of a correctly segmented picture that aids in the simple identification of healthy or contaminated samples. Plant disease detection systems have been demonstrated to function effectively using edge, threshold, location, and color-based segmentation approaches. As a result of the considerable color disparities between the infected leaf region and its native color, spot color-based segmentation emerges. In segmentation, determining a threshold value is critical.

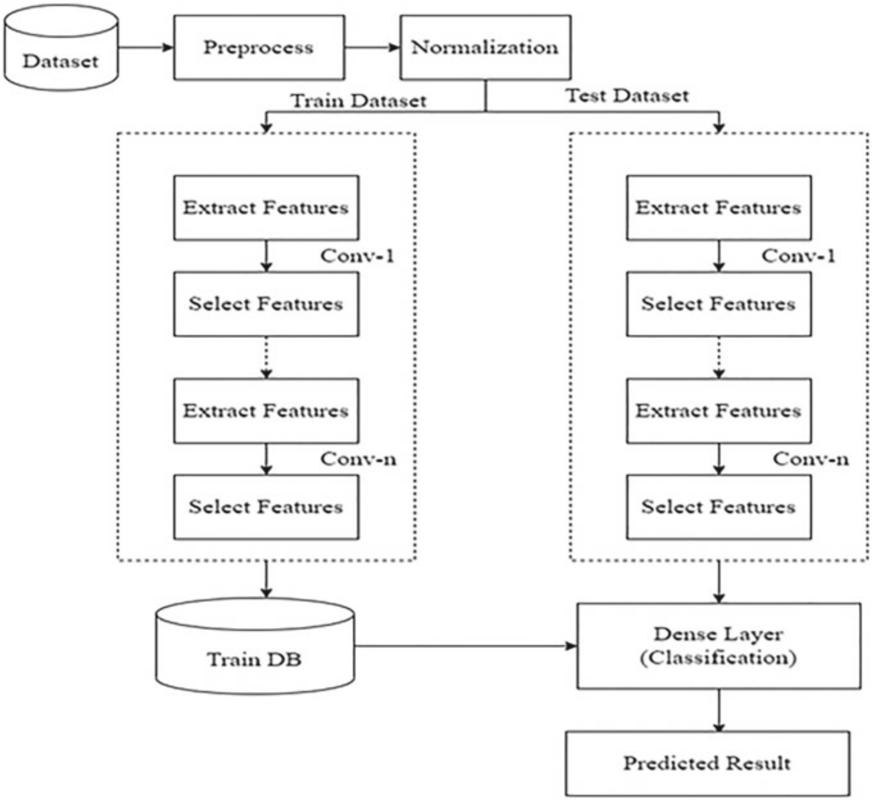


Fig. 1 Propose mCNN based plant disease detection and classification

Feature Extraction

Different features are available in images such as color, binary, histogram, autoencoder for identification of image. Shape and histograms are widely used to determine color. Texture may have properties including contrast, homogeneity, variance, and entropy. Diverse datasets need a variety of characteristics; however, texture has been recognized as the best feature for plant disease diagnosis. For feature extraction, a variety of approaches are utilized.

Classification or Recognition

In plant disease detection systems, classification is a critical component. The classification of test image either normal or abnormal has been defined in the testing phase. The training dataset was used for module training while the trained module was used for prediction of class label of the test dataset. The hybrid deep learning classification algorithm is what we used for detection of disease with cCNN and mCNN collaboration.

4 Algorithm Design

To implement this work, we design a new modified deep learning-based convolutional neural network classifier called mCNN. This algorithm is divided into two phases such as training and resting. The training module generates the rules for the entire module, while the testing phase validates disease detection and classification tests.

The (xit, yit) are the training set, and generate background knowledge for the entire train module. The primary function of the objective is

$$\min\ (\mathrm{W}) = \frac{\lambda}{2}\|\mathrm{W}\|^2 + f\big(W, \big(x_{i_t}, y_{i_t}\big)\big). \tag{1}$$

Secondly, Eq. 1 calculates the gradient distance which is demonstrated by using Eq. 2.

$$\Delta_t = \lambda W_t - \alpha t y_{i_t} x_{i_t}$$

$$\text{where}\, \alpha_t = \begin{cases} 1, & \text{if } y_{i_t}\langle W_t, x_{i_t}\rangle \\ 0, & \text{Otherwise} \end{cases} < 1\,. \tag{2}$$

The updated formula of matrix W is as follows.

where (λt) is the weighted matrix generated by Eq. 2 while Eq. 3 executes on behalf of itself.

$$W_{t+1} = \left(1 - \frac{1}{t}\right) W_t + y_{i_t} x_{i_t} \tag{3}$$

In practice, Formula (3) is used to find the minima or maxima by iteration.

The implementation of the proposed model for the training and testing phase are described in detail in the below section.

Execution of Training.

Input: Train_DB[] as training dataset, set of activation function AF[].

Output: Trained module in.PKL file for the entire splited dataset.

Step 1: Initialize both the algorithms Train_DB[], AF[], epoch_size.

Step 2: Extracted_Features_set &#xF0DF; Extract_Features(Train_DB[]).

Step 3: Selecetd_Features[] &#xF0DF; optimizer(Extracted_Features_set).

Step 4: Train.pkl &#xF0DF; Build_Classifier(Selecetd_Features[]).

Step 5: Return Train.pkl.

The above algorithm executes during the module training, in step 1 initialization was done with no. of epochs, convolutional layers etc. Step 2 describes an extract feature from training data and features are optimized in step 3. The classifier was trained in step 4 and it returned the trained module with.pkl file return in step 5.

Execution of Testing

Input: **Test_DB [] as testing instance set or individual patient record, Training Background Knowledge Train.pkl, User defines threshold Th.**

Output: Output_Map < Predicted_class_label, Similarity_weight > optimized instance recommend by classifier.

Step 1: Read all test data from *Test_Data[]* using the below function for validating to training rules, the data is normalized and transformed according to algorithm requirements

$$\text{test\_Feature(data)} = \sum_{\text{m}=1}^{\text{n}} (.\text{Attribute\_Set[A[m]} \ldots\ldots\ldots \text{A[n]}Test\_Data)$$

Step 2: select the features from extracted attributes set of test_Feature(data) and generate feature map using the below function.

Test_FeatureMap [t….….n] = $\sum_{\text{x}=1}^{\text{n}}$(t) &#xF0DF; test_Feature (x).

Test_FeatureMap[x] are the selected features in the pooling layer. The convolutional layer extracts the features from the input that passes to the pooling layer and those selected features are stored in *Test_FeatureMap.*

Step 3: Now read the entire taring dataset to build the hidden layer for classification of the entire test data in the sense layer,

$$\text{train\_Feature(data)} = \sum_{\text{m}=1}^{\text{n}} (.\text{Attribute\_Set[A[m]} \ldots\ldots\ldots \text{A[n]}Train\_Data)$$

Step 4: Generate the training map using the below function from the input dataset.

Train_FeatureMap [t….….n] = $\sum_{\text{x}=1}^{\text{n}}$(t) &#xF0DF; train_Feature (x).

Train_FeatureMap[t] is the hidden layer map that generates feature vector for building the hidden layer. Then evaluate the entire test instances with train data.

Step 5: After generating the feature map we calculate the similarity weight for all instances in the dense layer between selected features in the pooling layer

$$\text{Gen\_weight} = \text{CalcWeight(Test\_FeatureMap}|| \sum_{\text{i}=1}^{\text{n}} \text{Train\_FeatureMap[i])}$$

Step 6: *Return* .Gen_weight

The above algorithm describes the testing phase process of the proposed model called mCNN. In step 1 the test dataset was read with total attributes and features are extracted from test data in step 2. Similar process was done for training data in steps 3 and 4 respectively. The similarity calculation was done as like the dense layer in step 5. The generated weight returns by a similarity function that returns by step 6.

5 Results and Discussions

The python 3.6 with jupyter notebook opensource source framework was used for the proposed implementation. The RESNET-100 deep learning framework was utilized for implementation of CNN. A major modification was done in conventional CNN according to the algorithm given (Fig. 2).

Figure 3 displays mCNN classification accuracy using the plant image dataset; similar tests were used with different cross validations and the results are shown. According to this investigation, this experiment delivers the greatest average classification detection rate of 93.60% and 94.90% for mCNN utilizing Tanh.

In this experiment, we examined ReLU's accuracy rate using a plant images dataset; comparable tests were conducted with varied cross validation etc. According to this investigation, different cross validation accuracies of the classification for mCNN are 95.30% and 97.10%, respectively.

Above Fig. 4 describes the result with and without cross-validation. We have used a least of three hidden layers for the detection of the disease. Using this experiment, we conclude mCNN with sigmoid provides better detection accuracy than conventional ML algorithms.

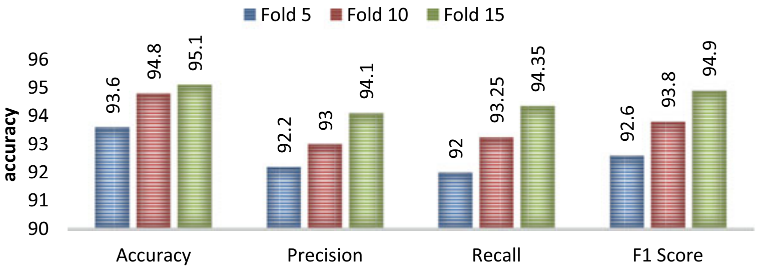


Fig. 2 Performance evaluation of proposed model using mCNN with sigmoid activation function

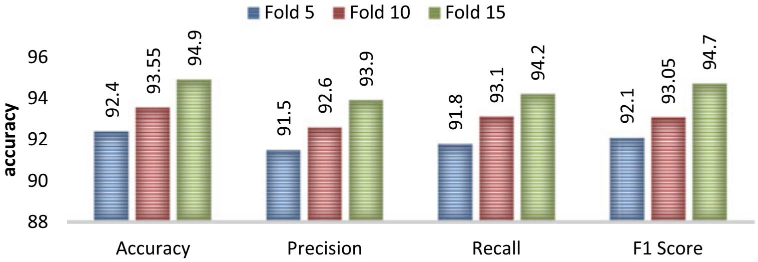


Fig. 3 Performance evaluation of proposed model using mCNN with TANH activation function

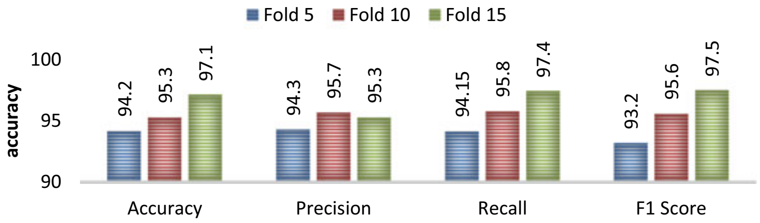


Fig. 4 Performance evaluation of proposed model using mCNN with RELU activation function

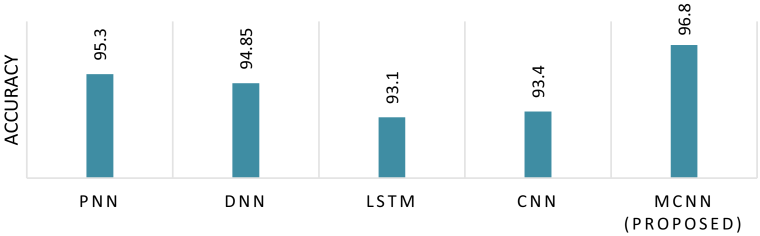


Fig. 5 Comparative analysis of proposed module with other deep learning module

Relative Analysis of Existing Deep Learning Algorithms

In another investigation, the probability of disease detection using supervised DL classification was detected. The system designates numerous evaluations among this research results and some existing systems results were calculated on the similar as well as multiple datasets.

Here, on Plant_Village dataset, no one has applied PNN, DNN, LSTM, CNN and the proposed mCNN based hybrid model. The accuracy achieved here is 96.80% which is better than other Deep Learning models. Figure 5 compares the proposed algorithms' classification accuracy to that of different known machine learning techniques. For data organization or classification, the most recent predicted sample employs a train and test data. The train dataset of the train model is made up of input function modules and their respective class labels. This learning set is used to create a categorization model that organizes the input data into appropriate template files or labels. The model is then validated using a test set derived from the class labels in the entire test dataset.

6 Conclusion

This paper describes a plant disease detection and classification using a modified DL framework called mCNN. The specialized Model was developed using DL and conventional ML algorithms using image processing to identify plant diseases using healthy and infected images of leaves. This can help the farmer for proper production of good quality crops. According to extensive experimental analysis the proposed DL model obtains higher results than conventional ML algorithms. DL is the new approach for detection and the classification of the plant disease in synthetic as well as real time dataset. We extract various heterogeneous features in the convolutional module and build a robust train module. The mCNN achieves 96.80% detection accuracy on heterogeneous plant image datasets, which is much higher than the conventional classification algorithms. Considering the heterogeneous plant dataset with various identification of diseases using hybrid DL algorithms will be the future task of the proposed research.

References

1. Savary S, Ficke A, Aubertot J-N, Hollier C (2012) Crop losses due to diseases and their implications for global food production losses and food security. Food Sec 4:519–537
2. Small family farmers, Family Farming Knowledge Platform, Food and Agriculture Organization of the United Nations. http://www.fao.org/family-farming/themes/small-family-farmers/en/
3. Gebbers R, Adamchuk VI (2010) Precision agriculture and food security. Science 327:828–831
4. Gewali UB, Monteiro ST, Saber E (2018) Machine learning based hyperspectral image analysis: a survey, pp 1–42. arXiv:1802.08701
5. Yao C, Zhang Y, Zhang Y, Liu H (2017) Application of convolutional neural network in classification of high resolution agricultural remote sensing images. In: The international archives of the photogrammetry, remote sensing and spatial information sciences, XLII-2/W7, pp 989–992
6. Rahnemoonfar M, Sheppard C (2017) Deep count: fruit counting based on deep simulated learning. Sensors 17:905
7. Lee H, Kwon H (2017) Going deeper with contextual CNN for hyperspectral image classification. IEEE Trans Image Process 26:4843–4855
8. Steen K, Christiansen P, Karstoft H, Jørgensen R (2016) Using deep learning to challenge safety standard for highly autonomous machines in agriculture. J Imag 2:6
9. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60:84–90
10. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp 1–9
11. Liu B, Zhang Y, He D, Li Y (2018) Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry 10:11
12. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 145:311–318
13. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7:1419
14. Arnal Barbedo JG (2019) Plant disease identification from individual lesions and spots using deep learning. Biosyst Eng 180:96–107
15. Patrício DI, Rieder R (2018) Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review. Comput Electron Agric 153:69–81
16. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems, vol 27. Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp 2672–2680

17. Chung C-L, Huang K-J, Chen S-Y, Lai M-H, Chen Y-C, Kuo Y-F (2016) Detecting Bakanae disease in rice seedlings by machine vision. Comput Electron Agric 121:404–411
18. Shrivastava S, Singh SK, Hooda DS (2017) Soybean plant foliar disease detection using image retrieval approaches. Multimed Tools Appl 76:26647–26674
19. Liu T, Chen W, Wu W, Sun C, Guo W, Zhu X (2016) Detection of aphids in wheat fields using a computer vision technique. Biosyst Eng 141:82–93
20. Too EC, Yujian L, Njuki S, Yingchun L (2019) A comparative study of fine-tuning deep learning models for plant disease identification. Comput Electron Agric 161:272–279
21. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE conference on computer vision and pattern recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp 580–587
22. Fuentes A, Yoon S, Kim SC, Park DS (2022) A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors 2017:17
23. Gutierrez A, Ansuategi A, Susperregi L, Tubío C, Ranki´c I, Lenža L (2019) A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. J Sens
24. Ramcharan A, McCloskey P, Baranowski K, Mbilinyi N, Mrisho L, Ndalahwa M, Legg J, Hughes DP (2019) A mobile-based deep learning model for cassava disease diagnosis. Front Plant Sci 10:272
25. Ji M, Zhang K, Wu Q, Deng Z (2020) Multi-label learning for crop leaf diseases recognition and severity estimation based on convolutional neural networks. Soft Comput 24:15327–15340
26. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the advances in neural information processing systems (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; pp 1097–1105
27. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: Proceedings of the European conference on computer vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp 740–755
28. Hughes D, Salathé M (2015) An open access repository of images on plant health to enable the development of mobile disease diagnostics, arXiv:1511.08060
29. Camargo A, Smith JS (2009) Image pattern classification for the identification of disease causing agents in plants. Comput Electron Agric 66:121–125

Jewel Beetle-Batteryless WSN Eco Monitoring System

Rajiv Iyer and Aarti Bakshi

Abstract This paper presents a system for monitoring forest fire using wireless sensor networks (WSN) and the Internet of Things (IoT). Forest fire is an important and unsolved ecological problem. Although many solutions are available in the literature and practice they have not been able to solve the problem effectively. Also, they have problems in terms of cost and ease of implementation. This paper deals with the design and development of a batteryless eco monitoring system and it is used on readily available energy sources. However, improvement of the performance is needed, therefore methods of harvesting energy around these sensors are implemented to extend the life of the battery or ideally provide an endless supply of energy to the sensor. To achieve this, we have designed 3 nodes working on the energy that will be provided from readily available energy from harvesting systems. These nodes with attached sensors are been used to collect the environmental data. The system is optimized in terms of reliability and sensitivity as compared to existing systems.

Keywords Arduino · Eco monitoring · Energy harvesting · Forest fire · IoT · Temperature sensor

1 Introduction

Forest fire is a problem that is yet not solved which is leading to major environmental changes all over the world. It is irreparably affecting the flora and fauna. Although a lot of techniques to detect forest fire are available in literature each has its own challenges in terms of cost, ease of implementation, complexity, and availability. Further battery-based systems require replacements which are difficult in forest environments and are not easily accessible to humans Thus there is a growing interest to harvest ambient energy for the operation of solutions designed for forest fire detection such as low-power wireless sensors. RF energy harvesting can be used to partially/fully

R. Iyer (✉) · A. Bakshi
Department of Electronics and Telecommunication Engineering, K.C College of Engineering and Management Studies and Research, Thane, India
e-mail: rajivkjs@gmail.com

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_18

supply the energy required for the operation of portable electronic devices such as wireless sensors, cell phones, Bluetooth devices, medical implants, and hearing aid devices. The work in this paper presents two energy harvesting techniques that can be deployed for forest fire detection systems. This paper also shows the three nodes developed using processors which are low energy, low cost, and have low processing power suitable for forest fire detection applications. In this paper, we will first review the work done previously in Sect. 2. Section 3 describes the proposed methodology and discussed the results obtained in Sect. 4. We conclude in Sect. 5.

2 Related Work

Zigbee-based methods are used in forest fire detection systems [1] to monitor temperature and humidity in the forest more promptly and accurately. The authors have highlighted the special benefits of data transmission security, network flexibility, and low cost and energy requirements for a forest fire monitoring system based on a Zigbee wireless sensor technology that they designed. The system's topological structure is a cluster tree adaption. A cluster tree structure is simpler to build than a reticular one, because the information flow requires less memory. The maximum range of Zigbee places restrictions on the system.

Also, problems of energy consumption, node location, and clock synchronization need to be addressed. As described in [2], a mechatronic evaluation of forest fire monitoring systems based on UAV is implemented. Drone-based systems are also used for these systems. The requirements are mapped to the mechatronic capabilities that these systems should possess. The supporting technology for these skills are succinctly described. The discrete Choquet integral is used to assess the architectural designs of these systems. Using drones, however, is not cost-effective in scale and faces regulatory issues in many countries.

In [3] a robotics-based mechanism is used. The authors created a drone-based unmanned aerial vehicle (UAV) and employed it for various robot mechanism applications. Magnetometer, temperature, and night vision cameras are all present. To detect the Earth's magnetic field, spot magnetic anomalies, and calculate the dipole moment of magnetic materials, magnetometers are frequently employed in geophysical surveys. A magnetic sensor can assist in the detection of landmines. In order to record or communicate temperature changes, a temperature detector detects the ambient temperature and turns the input line into electrical data. The image or tape-hung night vision camera uses both electrical and graphic camera detectors to permit moving objects within the monitored environment. Based on this module, the Unmanned Aerial Vehicle (UAV) will fly throughout the day, at night, in smoky areas, over uncharted territory, and in search of unknown activities in order to identify the forest fire for emergency purposes. The temperature will be detected, the forest fire will be located, and a response notification will be sent to the controller using a temperature sensor. However, using drones also has issues with the sensors that are mounted on them, especially magnetometers. In [4], a drone-enabled wireless sensor network (WSN) was used.

This article proposes the optimal weighted probability function, taking into account the remaining node energy, node spacing, and average energy of the network by the author. This feature helps optimize the clustering process for three levels of heterogeneity by minimizing the energy of the proposed network. The proposed method achieves a 29.45% and 52.48% increase in network life compared to existing algorithms. The reason for this significant increase in network life is to use the proposed features to select the most powerful node as the cluster head. However, the system is limited by flight time and complex mathematical modeling of the drone. In [5] authors establish a geostationary satellite-based forest fire monitoring system that can monitor areas of the Korean Peninsula 24 h a day for forest fire monitoring, and describe how to establish a forest fire monitoring system and use it in various ways. In order to establish a satellite-utilized forest fire monitoring system, they have concluded literature research, technical principles, forest fire monitoring means, and a satellite forest fire monitoring system. The satellite-utilized forest fire monitoring system can consist of one geostationary satellite equipped with infrared detection optical sensors and a ground processing station that processes data received from satellites to spread surveillance information. Forest fire monitoring satellites are located in the country's geostationary orbit and should be operated 24 h a day, 365 days a day. Forest fire monitoring technology is an infrared detection technology that can be used in national public interests such as forest fire monitoring and national security. It should be operated 24 h a day, and to satisfy this, it is efficient to establish a geostationary satellite-based forest fire monitoring satellite system. The satellite-based classical system lags continuous monitoring, requires complex processing and the information does not reach the ground in real-time though. In [6], the author reports on a new tag-based WSN that does not use chips and batteries, which fundamentally breaks all previous paradigms. Consisting of off-the-shelf components on a printed board, this WSN can acquire and transmit information without injecting or collecting DC power, while polling the node with a full-duplex transceiver design, thus the node itself. Not affected by self-interference. The WSN described does not require advanced and expensive manufacturing, but its unique parametric dynamic operation allows for superior sensitivity and dynamic range beyond what is achieved with on-chip sensors. Batteryless systems for forest fire detection are at a very nascent stage and are not available widely.

Thus there is a need for a low-cost solution for forest fire detection with a device that can harvest energy available through renewable sources which can be easily deployed in harsh conditions. This paper addresses the problems in existing systems and is easily deployable and cost-effective.

3 Proposed Methodology

In order to prove the concept of the proposed system, we implemented a processing unit with a temperature and humidity sensor on the main circuit board. We investigated the characteristics of the environmental monitoring applications and clarify the requirements for designing the batteryless WSN system. The Jewel Beetle (JB) system is placed in the forest area where it covers the maximum possible area for monitoring the changes in different physical parameters such as temperature, humidity, etc. When a sufficient amount of energy is charged in the energy storage unit, the Jewel Beetle node gets activated. In an event of an emergency, i.e. when the sensors detect an abnormal situation, the active node informs its control station. The JB node charges energy obtained from an energy conversion device to a storage unit. Since light sources will be found in nearly every place, we are using solar cells for the energy supply of the JB node to support a good variety of monitoring applications. Additionally, we are using the RF energy within the current implementation. The solar cell or RF unit is provided with an energy storage unit in order to store the obtained electrical power temporarily. Further, the energy kept within the energy storage unit is connected to the processing unit that is integrated with sensors to observe the environmental parameters. We are using IoT as a communication model. We have implemented three different nodes with reducing the size as well as cost.

3.1 RF Harvester

The energy harvester will be used by all three nodes being discussed in next section. Firstly we designed an antenna for harvesting RF energy as shown in Fig. 1. This circuit is further given to the doublers circuit shown in Fig. 2. To design an antenna we took a one-sided copper cladding PCB. Then the photoresist mask of that shape that we want to implement is applied to the PCB. For this photoresist, we used a permanent marker here. We applied a double coat of photoresist and let it dry. Then we took FeCl3 (ferric chloride) solution n dipped PCB in it. Shake it well for 10-20 min. By doing this the part without photo resists on PCB goes off. To remove the photoresist acetone solution is rubbed on it. In simple words, the marker part is removed and only the copper substrate remains. Now for the feed line, we need center feed. For this PCB is drilled at the center and from that, a copper wire is mounted and soldered. From this, we get to feed, and hence we got our desired antenna. The RF energy from this is passed through a doublers circuit and then given to the other devices of the system. To store this energy, we have used an electrolytic capacitor.

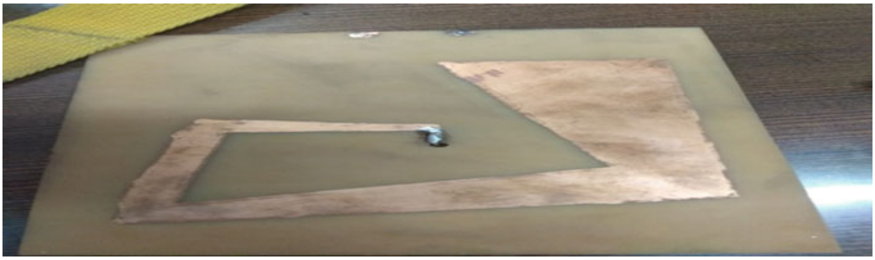


Fig. 1 RF Antenna developed for energy harvesting

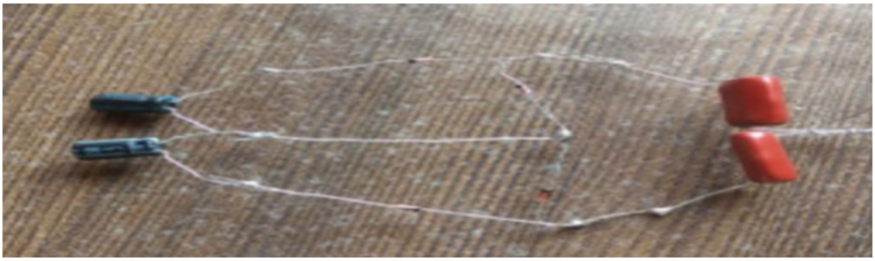


Fig. 2 Doublers circuit for RF energy harvesting

3.2 Proposed Models for Forest Fire Detection and Working

We have proposed three different nodes for forest fire detection (FFD); FFD 1, FFD2, and FFD3. Each model has a different hardware.

Node implementation using Arduino (FFD1).

In the above block diagram (see Fig. 3), the solar energy is extracted and converted into dc energy. This energy is stored in a rechargeable battery which charges through the battery charging rectifier circuit that gets the input energy through the solar. This battery charger is used to provide the power supply to the system. A voltage sensor is used to measure the battery voltage. Arduino UNO is used as a processor. Arduino controls the temperature and humidity sensor which is a DTH11 SENSOR and the Wi-Fi module which is ESP8266P. The program for processing is burnt in the Arduino using software ARDUINO IDE. Initially, Arduino is connected to the PC side web application using a Wi-Fi module. DTH11 sensor is used to measure the temperature and humidity of the environment. It will detect the temperature and humidity of the environment and Sensor data will be sent to the controller. If the temperature and humidity increase above a threshold then the controller will send the alert notification as fire is detected on the web application on the PC side through the Wi-Fi module. Here the database is created with date and time. This all is processed through a XAMP SERVER. As shown in Fig. 4, LCD (16 × 2) is used to display battery voltage status, temperature, and humidity. This node basically uses the RF energy as the resource. This energy is extracted using an antenna. The methodology used is explained in the flowchart as shown in Fig. 5.

Node implementation using Node MCU (FFD2)

As shown in Fig. 6, the Node MCU works as a processing unit as well as a Wi-Fi module. Moreover, it has an inbuilt web application so there is no need to use the XAMP server for the webpage. Figure 7 shows the hardware implementation of the FFD 2 node which has the Node MCU module with an inbuilt Wi-Fi module.

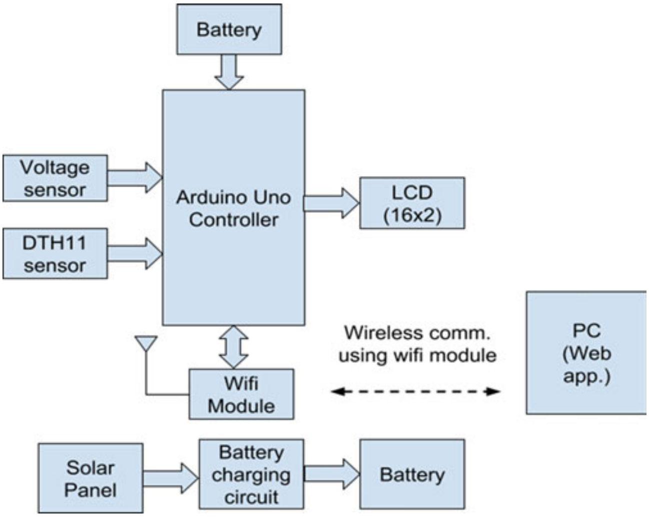


Fig. 3 Block diagram of node1

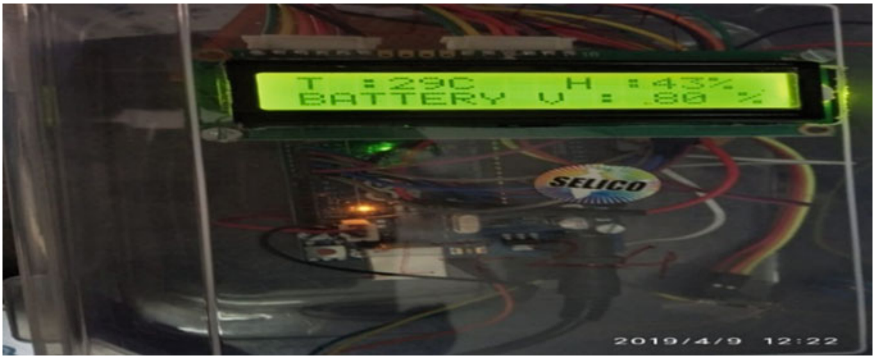


Fig. 4 LCD display

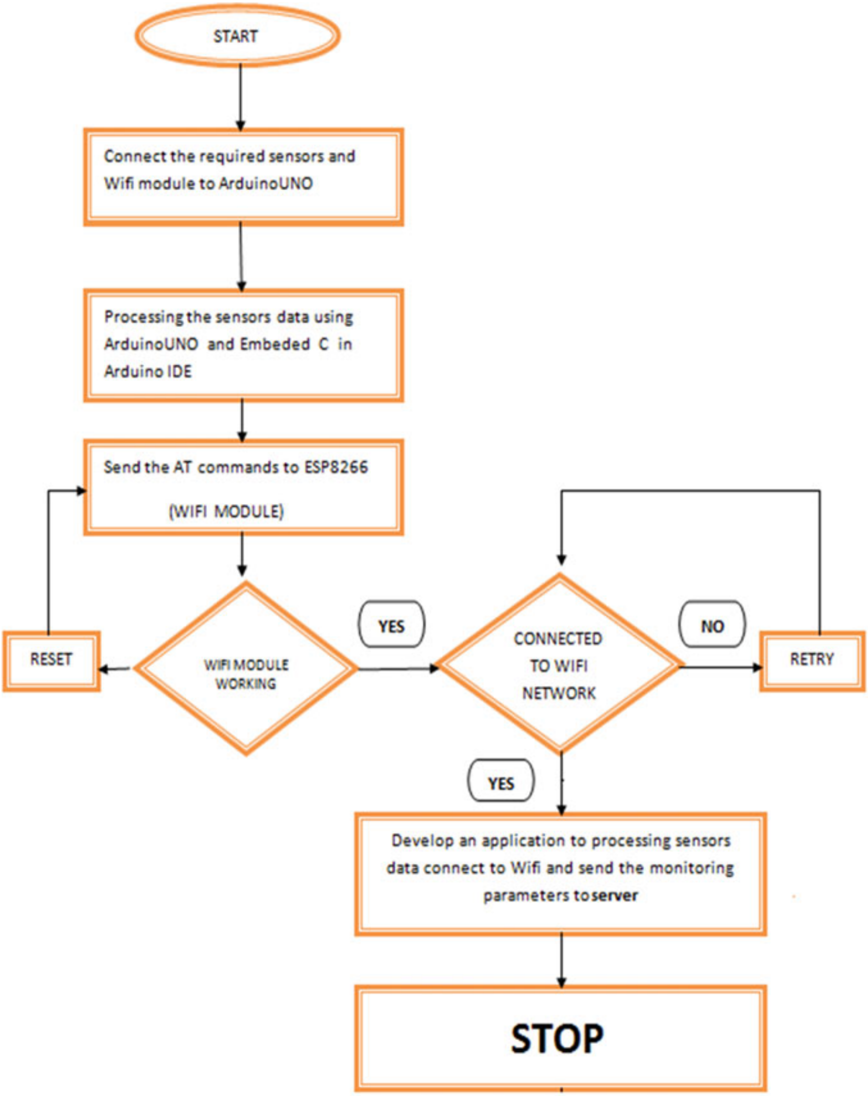

START
Connect the required sensors and Wifi module to ArduinoUNO
Processing the sensors data using ArduinoUNO and Embeded C in Arduino IDE
Send the AT commands to ESP8266 (WIFI MODULE)
RESET
WIFI MODULE WORKING
YES
CONNECTED TO WIFI NETWORK
NO
RETRY
YES
Develop an application to processing sensors data connect to Wifi and send the monitoring parameters to server
STOP

Fig. 5 Methodology of FFD node1

Node implementation using ESP01 (FFD3)

Here again in FFD3, as shown in Fig. 8, in the block diagram the RF energy is extracted using an antenna. The rest working is the same as in FFD2. The only difference is that Node MCU is replaced with the ESP01 as shown in Fig. 10 which works as a processor sending the data to the website as shown in flowchart Fig. 9. This reduces the hardware size as well as the cost much further.

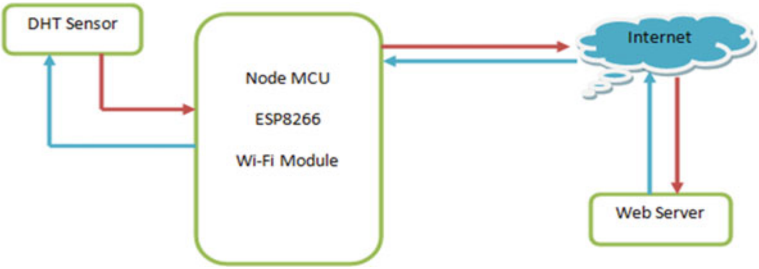


Fig. 6 Block diagram of node2

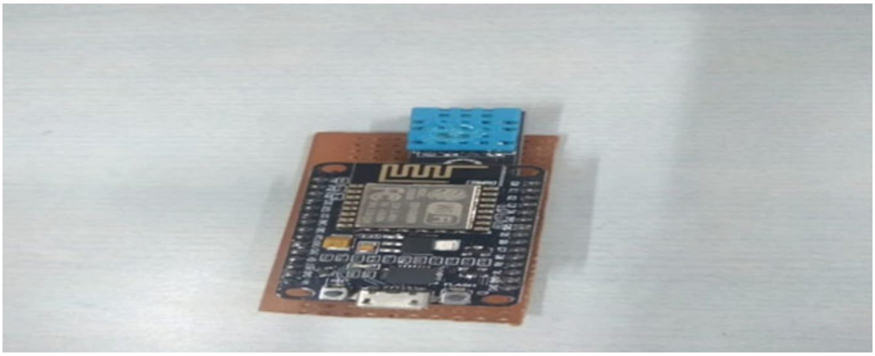


Fig. 7 Hardware implementation of FFD2 node

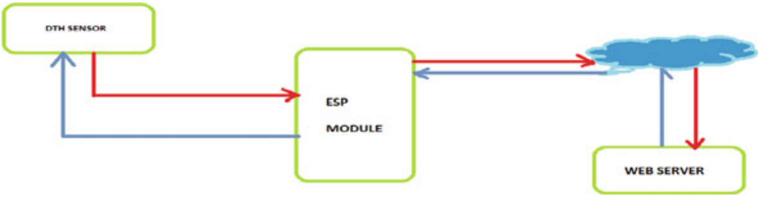


Fig. 8 Block diagram of node3

4 Result Analysis

4.1 Result for Node Implementation Using Arduino (FFD1)

The above figure shows how results will be displayed on the webpage. On the computer side web application, display of the DTH11 sensor status, voltage charging status and data log will be available as shown in Fig. 11.

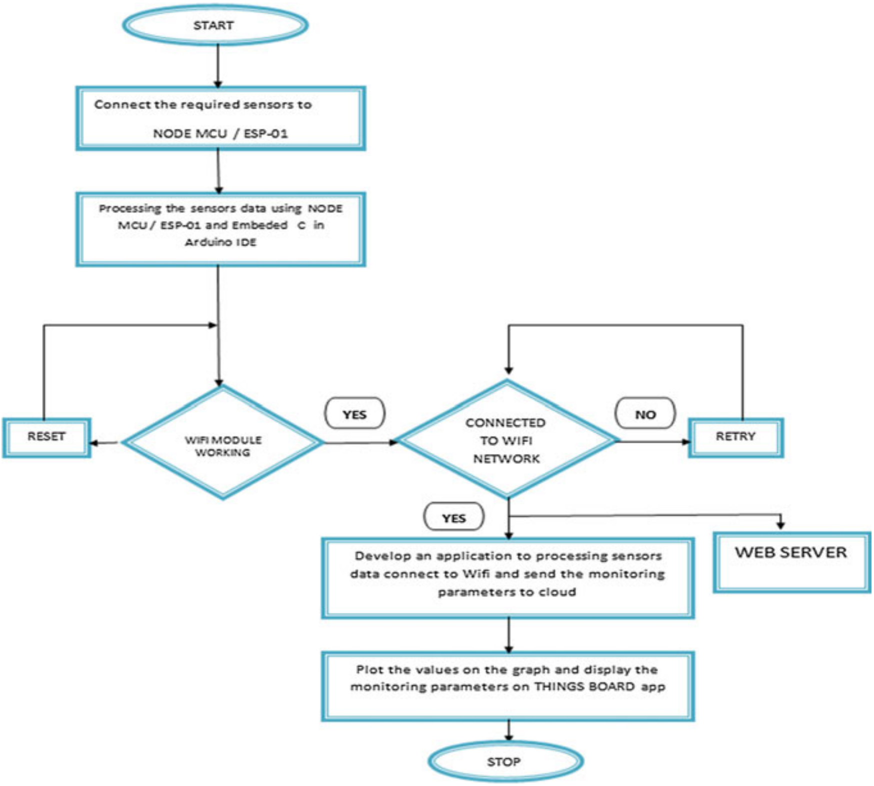

START
Connect the required sensors to NODE MCU / ESP-01
Processing the sensors data using NODE MCU / ESP-01 and Embeded C in Arduino IDE
RESET
WIFI MODULE WORKING
YES
CONNECTED TO WIFI NETWORK
NO
RETRY
YES
Develop an application to processing sensors data connect to Wifi and send the monitoring parameters to cloud
WEB SERVER
Plot the values on the graph and display the monitoring parameters on THINGS BOARD app
STOP

Fig. 9 Methodology of node2 and node3

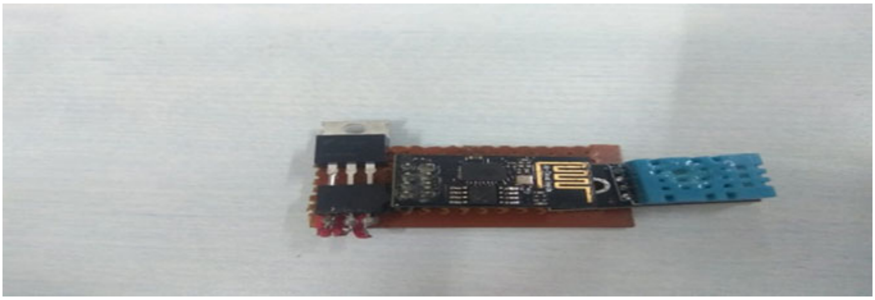
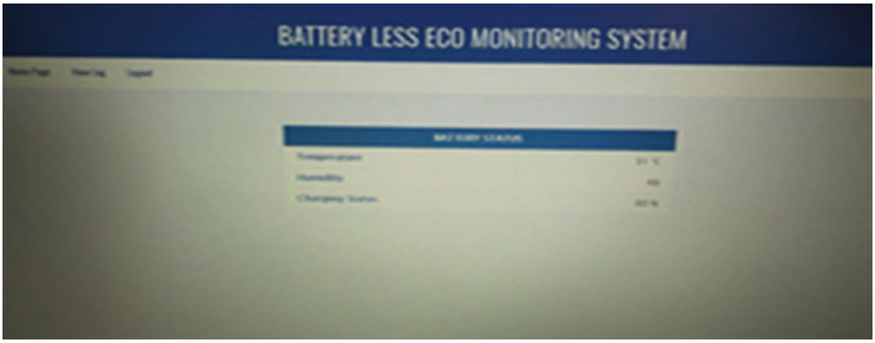


Fig. 10 Hardware implementation of node3



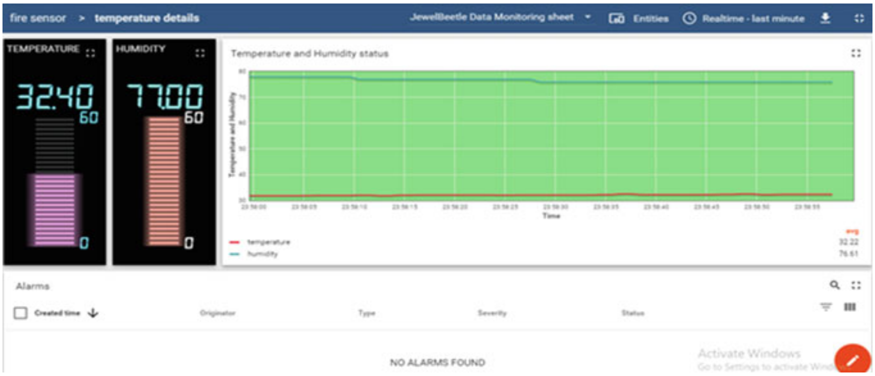
BATTERY STATUS

Temperature	[illegible]
Humidity	[illegible]
Charging Status	[illegible]

Fig. 11 Web Page displaying the value of temperature, humidity and charging voltage

4.2 Result for Node Implementation Using Node MCU (FFD2)

Using Things-board open source software the data is sent to the web side and monitored for real-time application. The data is not stored but it displays the present time values on the screen and displays the data in graphical form. Using the alarms function in the software, an alarm is created using the threshold value, which then alerts when the temperature exceeds the threshold value set. This system's main advantage is that it reduces programming at a major level. It also has a major reduction in the cost and the size. Figure 12 shows the results in normal conditions when there is no forest fire. Figure 13 shows the scenario when a forest fire is detected. As seen by comparing both the results it can be seen that the temperature and humidity value varies as soon as a forest fire is detected.

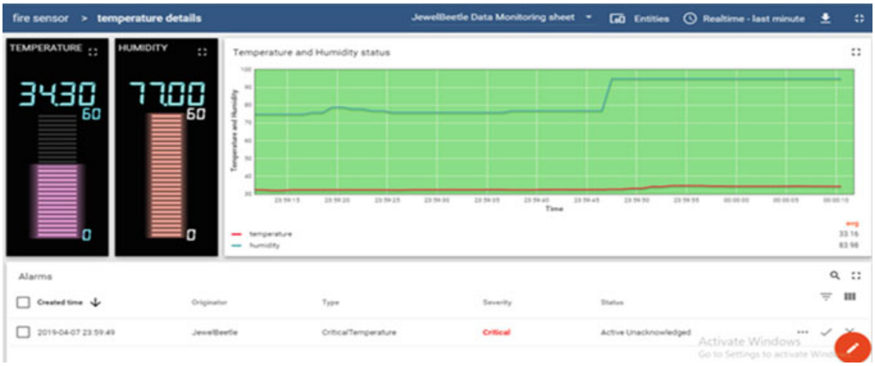


Alarms

□ Created time ↓	Originator	Type	Severity	Status

NO ALARMS FOUND

Fig. 12 Normal conditions



Alarms

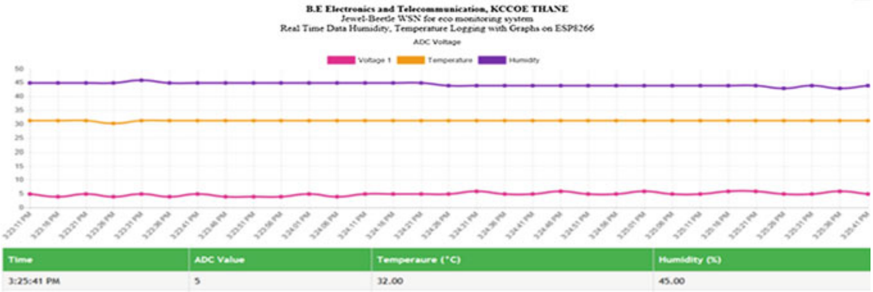
□ Created time ↓	Originator	Type	Severity	Status
□ 2019-04-07 23:59:49	JewelBeetle	CriticalTemperature	Critical	Active Unacknowledged

Fig. 13 Fire detection

4.3 Result for Node Implementation Using ESP01 (FFD3)

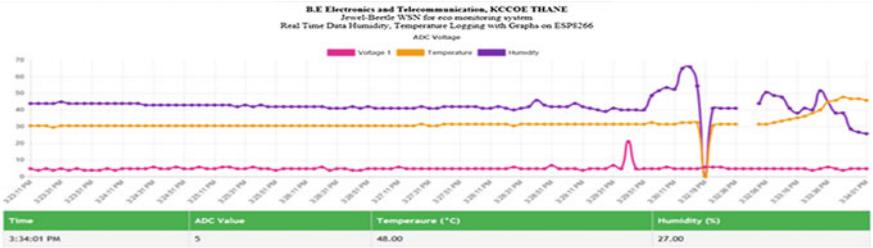
Using Things-board open source software the data is sent to the web side and monitored for real-time application. The data is not stored but it displays the present time values on the screen and displays the data in graphical form. Using the alarms function in the software, an alarm is created using the threshold value, which then alerts when the temperature exceeds the threshold value set. This system's main advantage is that it reduces programming at a major level. It also has a major reduction in the cost and the size. Figure 14 shows the results in normal conditions when there is no forest fire whereas Fig. 15 shows the scenario when a forest fire is detected. As seen by comparing both the results it can be seen that the temperature and humidity value varies as soon as a forest fire is detected. It is showing the ADC voltage change as well as compared to node 2 where only temperature and humidity were displayed.

The comparative analysis yields that our proposed Jewel Beetle system's sensitivity and resolution are improved compared to satellite and drone-based systems [2–6]. As our proposed system is an on-site system and it can be deployed in the forest permanently.


B.E Electronics and Telecommunication, KCCOE THANE
Jewel-Beetle WSN for eco monitoring system
Real Time Data Humidity, Temperature Logging with Graphs on ESP8266

Time	ADC Value	Temperaure (°C)	Humidity (%)
3:25:41 PM	5	32.00	45.00

Fig. 14 Normal conditions



B.E Electronics and Telecommunication, KCCOE THANE
Jewel-Beetle WSN for eco monitoring system
Real Time Data Humidity, Temperature Logging with Graphs on ESP8266

Time	ADC Value	Temperaure (°C)	Humidity (%)
3:34:01 PM	5	48.00	27.00

Fig. 15 Fire detection

5 Conclusion

The Batteryless eco monitoring system is a specialized system for supporting typical scenarios of environmental monitoring applications. For environmental monitoring the IoT-based solutions were designed, developed, and analyzed. The analysis of the implementations revealed the fact that Wi-Fi technologies are suited for monitoring applications. As expected, Wi-Fi consumes more energy but enables the development of solutions with a reduced total cost of ownership through the use of the existing infrastructure. It is a system that can monitor the temperature and humidity level using Arduino controller that helps to analyze the various patterns in the environmental parameters while IoT is proposed which can help in data sending on web application side on PC and accordingly notifies. By the use of DTH11 sensors, the temperature and humidity can be sensed and according to the sensor status, a fire alert can be given. The Arduino controller serves as the heart of this module which controls the entire process. The Wi-Fi module connects the whole process to the internet. The proposed solution in this paper is low-cost and easily deployable in a forest environment.

References

1. Zhang J, Li W, Han N (2008) Forest fire detection system based on a ZigBee wireless sensor network. In: Frontiers of forestry in China, vol 3. Springer, pp 369–374
2. Moulianitis VC, Thanellas G, Xanthopoulos N, Aspragathos NA (2018) Evaluation of UAV based schemes for forest fire monitoring. In: Advances in service and industrial robotics RAAD 2018. Mechanisms and machine science, vol 67. Springer, Cham, pp 143–150
3. Sivabalaselvamani D, Selvakarthi D, Rahunathan D, Munish M, SaravanaKumar R, Sruthi S (2022) Forest fire and landmines identification with the support of drones surveillance for better environmental protection: a survey. In: 4th international conference on smart systems and inventive technology (ICSSIT) proceedings, IEEE, pp 1480–1485
4. Singh S, Malik A, Kumar R, Singh PK (2021) A proficient data gathering technique for unmanned aerial vehicle-enabled heterogeneous wireless sensor networks. Int J Commun Syst 34(16)
5. Park BS, Cho IJ, Lim JH, Kim IB (2021) Forest fire monitoring system using satellite. J Converg Inf Technol 11(11):143–150
6. Hussein HME, Rinaldi M, Onabajo M (2021) A chip-less and battery-less sub harmonic tag for wireless sensing with parametrically enhanced sensitivity and dynamic range. Sci Rep 11:3782

An Interactive Platform for Farmers with Plant Disease Recognition Using Deep Learning Approach

I. Aryan Murthy, Niraj Ashish Mhatre, Rohit Vasudev Oroskar, Surya Voriganti, Keerti Kharatmol, and Poulami Das

Abstract India's primary sector is mainly contributed by the agriculture industry. The GDP from agriculture in India increased to 6630 billion INR in the fourth quarter of 2021 from 4076 billion INR in the third quarter of 2021.

Due to the varied conditions of the subcontinent, the farmers may face different problems affecting the crop yield which ultimately affects the economy. Also, in case of the crops the decisions need to be immediate since if the crops are affected by a disease, then it can spread over the whole field affecting other crops.

In order to help the farmers, come up with solutions to these problems, it has been aimed to create a platform on which the farmers can post their queries and other individuals can respond to those questions with answers. A deep learning approach has also been used to detect plant diseases. This will help the farmers to detect the disease of 14 different plant species with 38 sub-classes and also to post their queries in the proposed platform.

Keywords Convolution neural network · Gross domestic product · Indian rupee · Plant disease recognition · Interactive platform · Residual network

I. A. Murthy · N. A. Mhatre · R. V. Oroskar · K. Kharatmol · P. Das (✉)
Department of Computer Engineering, K.C. College of Engineering and Management Studies and Research, Mumbai 400603, India
e-mail: dr.poulamidas.cse@gmail.com

I. A. Murthy
e-mail: iaryan@kccemsr.edu.in

N. A. Mhatre
e-mail: mhatreniraj@kccemsr.edu.in

R. V. Oroskar
e-mail: oroskarrohit@kccemsr.edu.in

K. Kharatmol
e-mail: keerti.kharatmol@kccemsr.edu.in

S. Voriganti
Automation and Robotics Specialist, GlobalMed Inc, Trenton K8V 5R5, Canada

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*,
Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_19

1 Introduction

Around the globe, there are approximately over half a billion farms. The agricultural sector is notably the prime industry for the Indian economy, also being a huge employer. About 60% of the country's populace works in the agro-based industry, contributing up to 18% to the nation's GDP.

Climate emergency, decline of pollinators, plant malady, etc. threaten food security. Diseases of plants pose a threat to food safety and additionally have cataclysmic adversities for farmers. The prevention of plant diseases costs time, cost and different other resources. Identifying these diseases is the first step toward its prevention. Not all diseases have visible symptoms. Detecting these diseases requires technology. Fungicides, disease-specific chemicals, and pesticides are some of the applications. In this paper, an interactive platform has been proposed where the farmers can upload the images of the plants and the diseases of the plants get detected by deep learning approach. Parallel to that, farmers can upload their queries and get answered by the experts as well as other farmers who have already faced the same problem.

For plant disease recognition, different adaptive versions of Convolution Neural Network (CNN) have been used. CNN is one of the most influential deep learning techniques used for pattern recognition of hefty data sets [1]. There is promising evidence that CNN can detect these diseases. Recognition can be achieved by using various classification architectures in deep learning. Transfer learning models include AlexNet [2], AlexNetOWTBn [2], GoogLeNet [3], and Overfeat [4]. They piled up many convolutional layers. Deep learning networks present difficulties, including degradation and vanishing gradient problems.

The paper gives a detailed analysis of the methodology and ResNet50 in Sect. 3, the proposed design of the website along with an explanation of the dataset used in this research is explained. Results and discussions mentioned provides a comparative analysis of different CNN models and concludes why ResNet50 is best among all the other models.

2 Related Work

In the recent years, researchers have discussed plant diseases and found solutions using many deep learning techniques. They poured their thoughts on how can plant diseases be detected and how to minimize it.

In 2018, Sardogan et al. [5] derived a model and 'learning vector quantization' theorem to successfully detect disease in tomato leaves. In 2019, Suresh et al. [6] made use of deep learning techniques along with inception_v3 PyTorch framework for good accurate results. In reference number [7], limited textured feature, for example, homogeneity and exhaustion were derived. The aim of the presentation was to recognize the disease in maize leaves.

In 2017, Pawara et al. [8] analyzed different feature description technologies with the help of CNN models. The comparative analysis included HOG-BOW blended with Support Vector Machine and Multi-layer perceptron classifier and Histogram of Oriented Gradients based features fused with K-Nearest Neighbors. The comparison of these models was done with the help of GoogleNet and AlexNet. In 2012, Chaudhary [9] made use of different color-based technologies to crop a fixed region of recognition in the pictographs of crop leaves. In this research, the color models which were used are CIELB, HIS, etc. Fujita et al. [10] derived a 4-layer CNN model that can detect seven different types of plant diseases and also included cucumber leaves (healthy). The model accuracy was up to 83%.

Khirade and Patil [11] used few segmentation technologies and derived conclusions from plant leaf pictures and implemented CNN along with back propagation. Bashish et al. [12] made use of k-means clustering and a pre-trained neural network-based model for stem and leaves disease recognition. Sankaran et al. [13] derived a swift and handy plant health observation sensor. For observing plant diseases, they showcased numerous technologies that were used to spot the crop diseases. Waldchen and Mader [14] used computer vision technologies to review different plant diseases.

In this research, a CNN model is derived by pre-trained ResNet50 architecture. The extra layers in the model majorly profits from the feature extraction procedure. Along with the process, fine tuning is performed to increase accuracy of detecting diseases. The dataset in the research includes 38 sub-classes of different leaf diseases along with the pictures of healthy crops.

3 Methodology

3.1 Convolution Neural Network

Inspired from the negonitron, the initial work on Convolutional Networks was introduced in 1990s. Yann LeCun et al., described the features of a CNN model in his paper [15]. Convolutional Neural Network is a deep learning neural network devised for filtering data in the form of images. It is composed of multiple layers and computes the data in a grid-like manner. These are utilized in computer vision and have been prominent in many visual applications like image stratification and text recognition. CNN is peculiar at pattern recognition taking input as image and can operate directly on raw image. CNN is differentiated from rest of the neural networks by their improved efficiency with image, speech and audio inputs [15].

RestNet – 50.

ResNet50 is a deep learning CNN model which is up to 50 layers. The full form for ResNet50 is Residual Network in which 50 indicates 50 layers. It is one of the many neural network models applied in many computer vision tasks. ResNet50 allows us

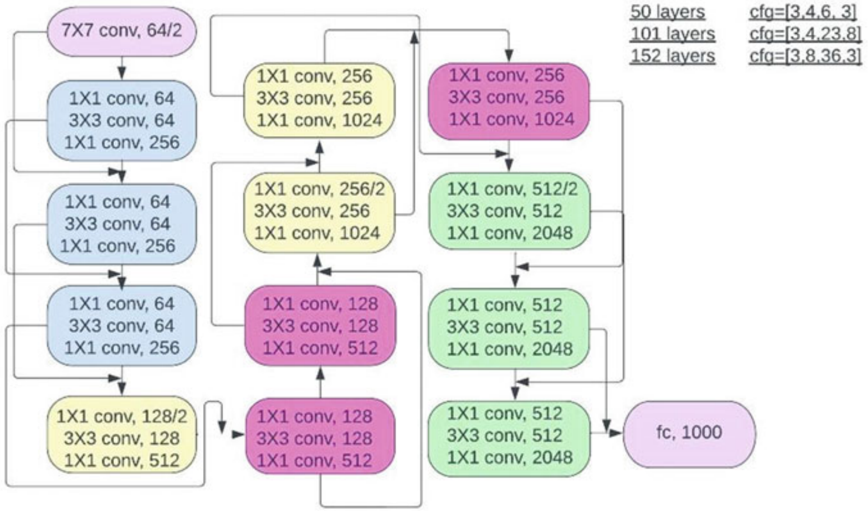


Fig. 1 The ResNet50 Architecture

to train deep models of up to 150 + layers. This model was created by Xiangyu Zhang, Kaiming He, Jian Sun, and Shaoqing Ren in their research paper dedicated to computer vision, ‘Deep Residual Learning for Image Recognition’ in the year 2015.

CNNs have a drawback of ‘Vanishing-Gradient Problem’. At the time of backpropagation, merit of gradient decreases gradually, thus change in weights of the model comes very rarely. To overcome this, ResNet is used.

ResNet50 has residual networks which are 50 layers deep. Figure 1 shows the architecture of residual network. It has groups of similar layers which is shown by different pixel in Fig. 1. Identified blocks are indicated by curved lines, which indicate that previous layers will be used in subsequent group. The main benefit of ResNet50 is that it minimizes the vanishing gradient problem.

From Fig. 1, it is observed that first group has 64 filters with a maximum kernel of 7 × 7 size, led by a maximum pool group of 3 × 3 size. The first group of residual network layers comprises three similar blocks. Similarly, 2nd group, 3rd group and 4th group consist of 4 similar blocks, 4 similar blocks and 3 similar blocks respectively. There are 38 connected layers used for the classification process. In the research, we do not use the connected layers because the pre-trained model is used.

4 The Proposed Design

The main aim of the research is to provide farmers a user interface in which they can upload images of diseased plants and get solutions either through experts registered on the website or through plant disease recognition. The interface will be easily approachable and farmers will be able to operate it efficiently. For a major implementation, plant disease prediction system using deep learning approach is used by which the farmer without asking any query can directly use and identify the problem on his own. The platform will help farmers to upload the images in a thread format so that any user can see the problem posted by the farmer and can write down solutions in comment format. Even in the future, if anyone faces same problem then, with one search they can find previous solutions posted on the platform. This will make the work much easier.

4.1 Technology Used

Django. It is a web development framework that enhances fast development and clean design. Django's main aim is to make the database-driven websites simple while developing it. The framework emphasizes reusability of components, less amount of code, fast development, and the principle of not repeating itself again. Django only relies on python.

HTML. HTML is a language use to create display design for the web browser. It enables users to provide the sections a proper alignment.

CSS. It is a style sheet language that enables the description of the document created in HTML. CSS is made to divide content and presentation, including colors, fonts.

SQLITE. It is a database engine used to store all the libraries and data.

4.2 Use Case of the Platform

The use case diagram in Fig. 2 illustrates the features provided in the platform for the user. Registration, Question and answer, upvotes and downvotes, notifications and plant disease prediction system are the major features of the platform. The prediction system has been linked to the website which can be redirected by a single click which makes its use much easier.

4.3 Data Flow

Figure 3 illustrates the data flow diagram level 0 of the platform and how data is handled by the website in Fig. 4.

The website mainly comprises of 5 major external entities namely, Farmer, plant expert, coordinator, prediction system, maintenance. The farmer entity posts the queries and can use the prediction system. Coordinator informs about the regular

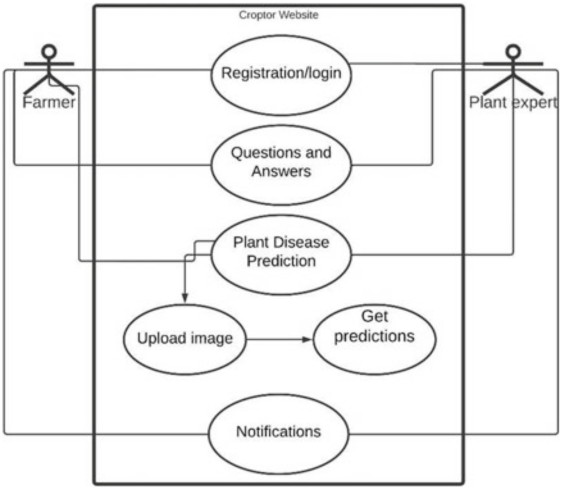


Fig. 2 Use Case Diagram of the platform

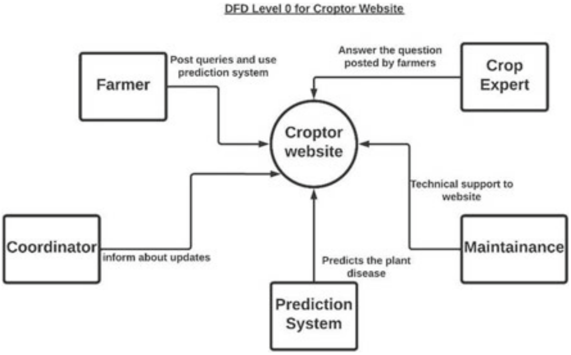

DFD Level 0 for Croptor Website
Farmer
Post queries and use prediction system
Answer the question posted by farmers
Crop Expert
Croptor website
Technical support to website
Coordinator
inform about updates
Predicts the plant disease
Maintainance
Prediction System

Fig. 3 Data flow diagram level 0

updated, maintenance provides technical support to the website. The prediction system provides predictions of up to 14 different plant species. The external entity offers support to the questions posted on the website by providing technical answers for the same.

The entire deep learning model making use of CNN for plant disease recognition is elaborated above. The steps involved in the working flow of model is displayed in Fig. 5. The deep learning CNN model initiates by training images further, pre-processing them, applying augmentation, utilization of pre-trained ResNet50 weights, optimization of model parameters [18]. The test was then carried out with an extensive conclusive analysis.

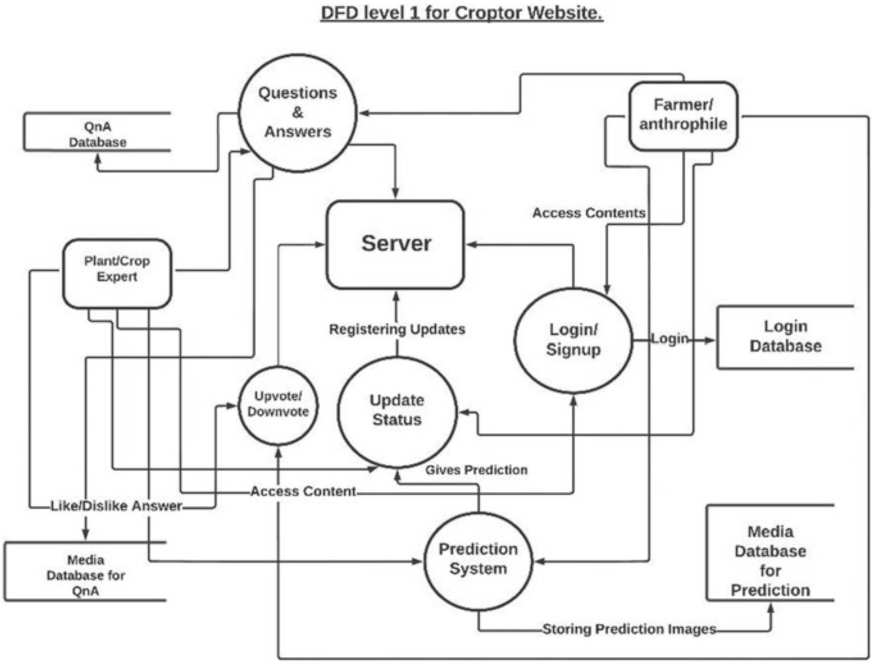
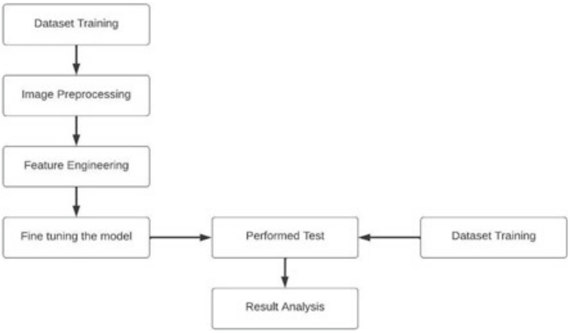


Fig. 4 Data flow diagram level 1

Fig. 5 Work-flow of deep learning model using ResNet50 architecture



5 Results and Discussions

5.1 The Dataset

The plant Village dataset taken into consideration for the research was sourced from ‘spMohanty’s GitHub repository’ [16]. The dataset is composed of images of healthy crops and plants diagnosed with disease. The suggested model was trained by different stratifications of leaves for detection. The set is composed of 54,309 images of 14 different plants namely, blueberry, apple, grape, cherry, maize, corn, tomato, soybean, orange, raspberry, squash, strawberry, bell pepper, potato. It is composed of images of up to seventeen fungal diseases, two viral-diseases, two molds diseases, and one disease caused by mites. There are 12 plant species images which have healthy leaves which do not show visible manifestation of a disease (Fig. 6).

Training of the model was done such that they can distinguish between group of plant diseases training datasets and validation datasets. The dataset was divided into 80% training dataset and 20% validation dataset from the color images provided in the dataset. In Fig. 7, it can be observed that the training and validation dataset length is divided into 80% and 20% respectively.

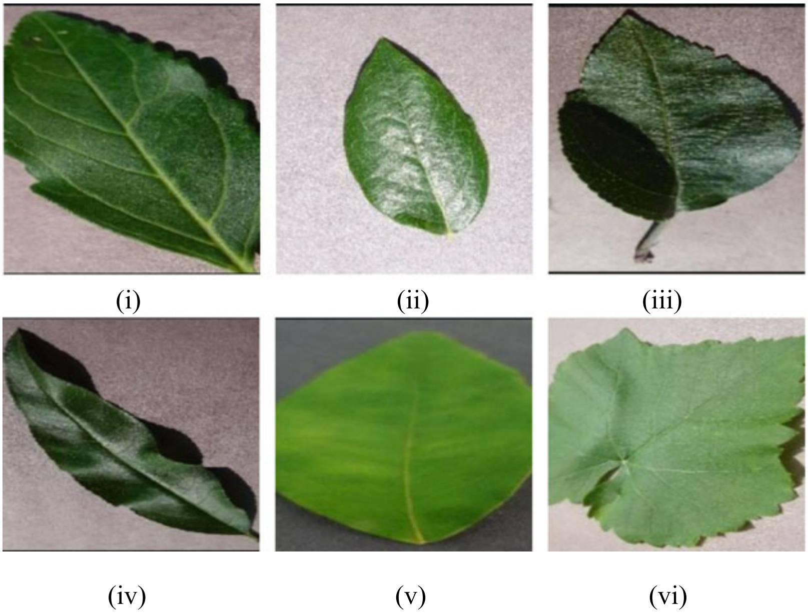
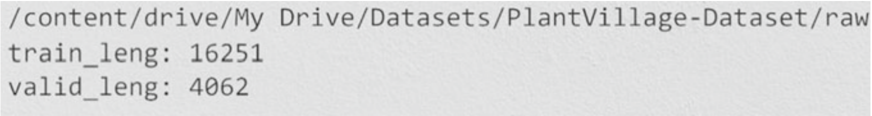


Fig. 6 Samples of new plant disease dataset: (i) cherry healthy, (ii) blueberry healthy, (iii) apple healthy, (iv) peach, (v) Orange, (vi) grape healthy



```
/content/drive/My Drive/Datasets/PlantVillage-Dataset/raw
train_leng: 16251
valid_leng: 4062
```

Fig. 7 The dataset is divided into validation and training dataset [17]

5.2 Image Pre-processing

Image pre-processing enhances the data required for image stratification. In pre-processing technique, there exists many dimensional modifications of images like scaling, translation, and rotation. In pre-processing, resolution of all the images is adjusted into 224 × 224 pixels. It is necessary to make sure that every image have identical resolution. For finding image easily, it is required to label or stratify by a keyword-search. Meanwhile, all transcribed images were eliminated from the dataset. The image dataset is classified under a keyword which makes the detection task easy. Also, due to same size resolved, the image searching and disease classification becomes faster.

5.3 Pretrained ResNet50

Instead of building the whole model from scratch for the similar problem, a pretrained model of ResNet50 was used. In the research with suggested model, every image was rescaled into 224 × 224 pixels in image pre-processing. The pre-trained ResNet50 model weights were used for better accuracy. Stochastic gradient descent (SGD) optimizer, and batch size of 04 was used for better accuracy. In the deep learning model, ‘learning rate’ was calibrated to 0.001, and ‘momentum and decay’ was calibrated to default value. By this, it was possible to increment the number of sub-classes where most of the former works includes low amount of sub-classes [18]. After that, the pretrained ResNet50 framework was applied to the classification on the dataset and then examined the capability of the model with the help of test images. Comparative analysis was performed by changing ResNet50 model weights.

5.4 Test Phase

Many tests were performed in different test setups to analyze the accuracy of the designed model. Many network variables are updated through the instruction given to the CNN model. The total dataset was split into 20% for validation purpose and 80% for training purpose as illustrated in Fig. 7. Then, the dataset was observed using the ResNet50 model [17]. No changes were assumed in the pretrained model and kept reserved the model as Resnet50 itself for better precision.

Table 1 Plant disease prediction system parameters

Parameter	Value
Validation steps	1
Batch-size	04
Steps-per-epoch	550
Optimizer	25
Epochs	SGD (stochastic gradient descent)
Learning rate	0.001
Decay	Default
Momentum	Default

5.5 Fine-Tuning of the Model for Better Accuracy

Adjusting is utilized for enhancing the productivity of a method used. It updates minor modifications to improvise the output required. The refinement procedure is very important that minor variation influences change in the training phase highly in regard to the computation time required, the convergence rate and the fining units used. This procedure of adjusting was performed over multiple times to increase the precision of the model. The variables are enumerated in Table 1.

The PyTorch model started training along with the trained dataset composed of both original images and those gained from augmentation. Then validation is carried out to generalize the model for better accuracy. Figure 8 depicts a fine slope of suggested network in the trained and validation process respectively. Even though there is a low-maxima in the validity curve, it displays 1.00 accuracy of validation for majority of the remaining curve.

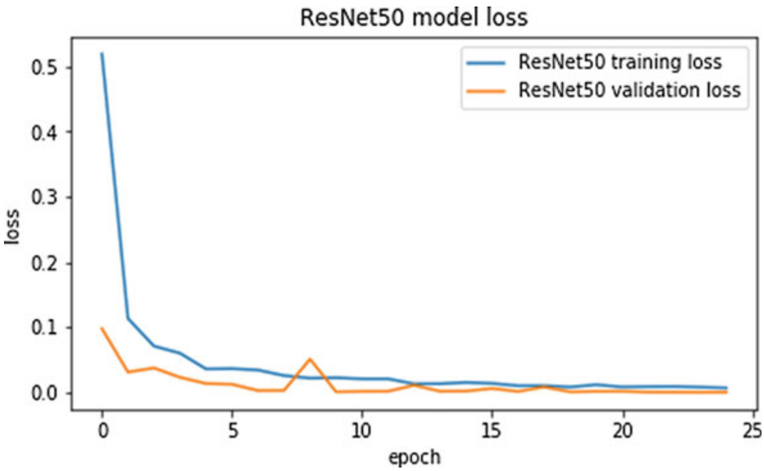


Fig. 8 Distribution of accurate results in training process [17]

Figure 9 depicts the scattering of losses in both the validation and training loss in regard to number of epochs in validation and training process. In Fig. 9, the curve illustrates the number of images that were precisely identified in the validation phase. Initially, the loss was high, but as the number of epochs increases, loss decreases gradually. Hence, epoch is inversely proportional to the training and validation loss.

Figure 10 illustrates the website homepage and Fig. 11. shows the webpage of plant diseases recognition.

Lastly, samples were applied in the test phase from 14 crop species and 38 subclasses. Sample detected images are depicted in Fig. 12. During the test, the outcome of the suggested models came with 100% precision along with two other probabilities of what the species might indicate to be.

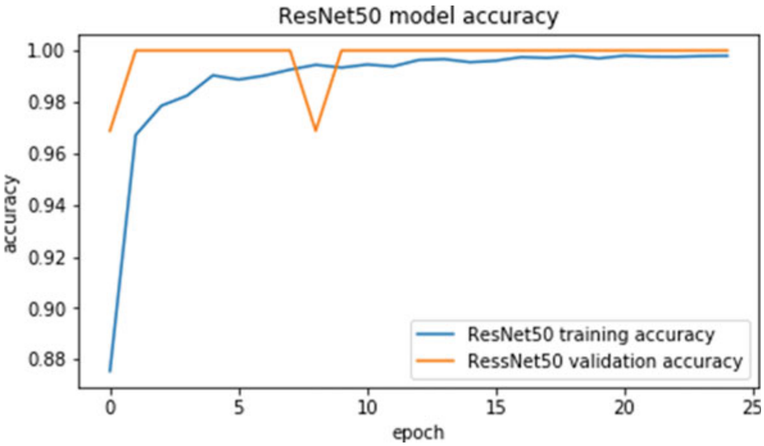
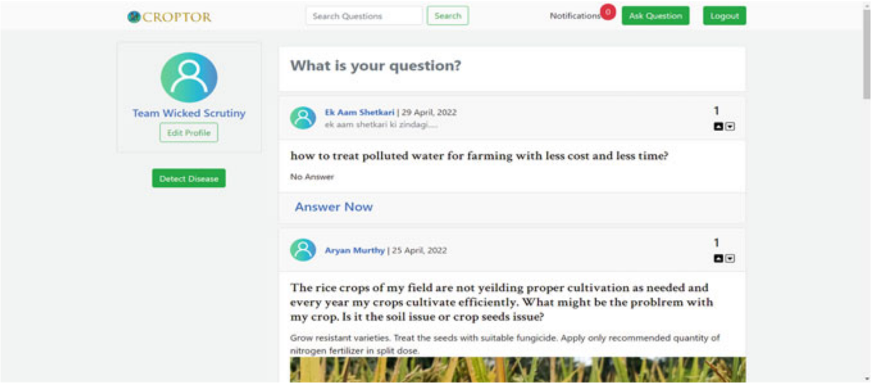


Fig. 9 Loss distributed along with epoch, 'validation phase' [17]



What is your question?

Ek Aam Shetkari | 29 April, 2022
ek aam shetkari ki zindagi.....

how to treat polluted water for farming with less cost and less time?

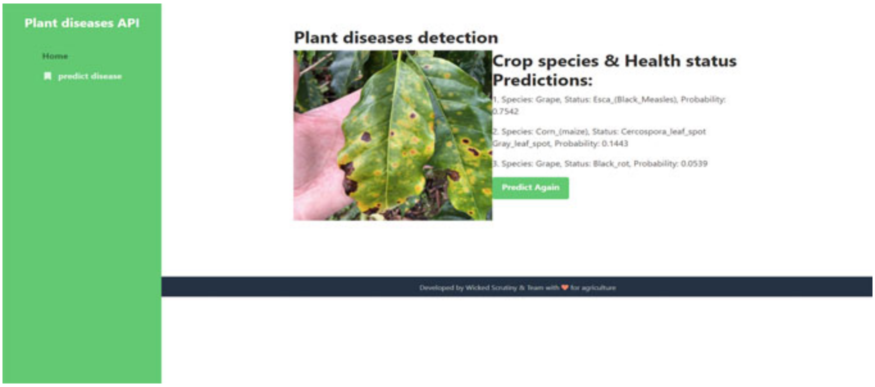
No Answer

Aryan Murthy | 25 April, 2022

The rice crops of my field are not yeilding proper cultivation as needed and every year my crops cultivate efficiently. What might be the problrem with my crop. Is it the soil issue or crop seeds issue?

Grow resistant varieties. Treat the seeds with suitable fungicide. Apply only recommended quantity of nitrogen fertilizer in split dose.

Fig. 10 User Interface of the webpage



Plant diseases detection

Crop species & Health status Predictions:

1. Species: Grape, Status: Esca_(Black_Measles), Probability: 0.7542
2. Species: Corn_(maize), Status: Cercospora_leaf_spot Gray_leaf_spot, Probability: 0.1443
3. Species: Grape, Status: Black_rot, Probability: 0.0539

Developed by Wicked Scrutiny & Team with ❤ for agriculture

Fig. 11 User Interface of the prediction system

The intersection generated in the training phase is higher in the deep learning model. The validation curve also accompany the training curve stipulating even convergence of the CNN model. The model is swift and productive in training. It can also foretell the test images precisely with less amount of time. For better conclusive report, the accuracies of all the models taken into consideration were determined. To achieve this, tests were carried out by varying the number of epoch and tried to maintain a consistent precision. The outcome is summarized in Table 2.

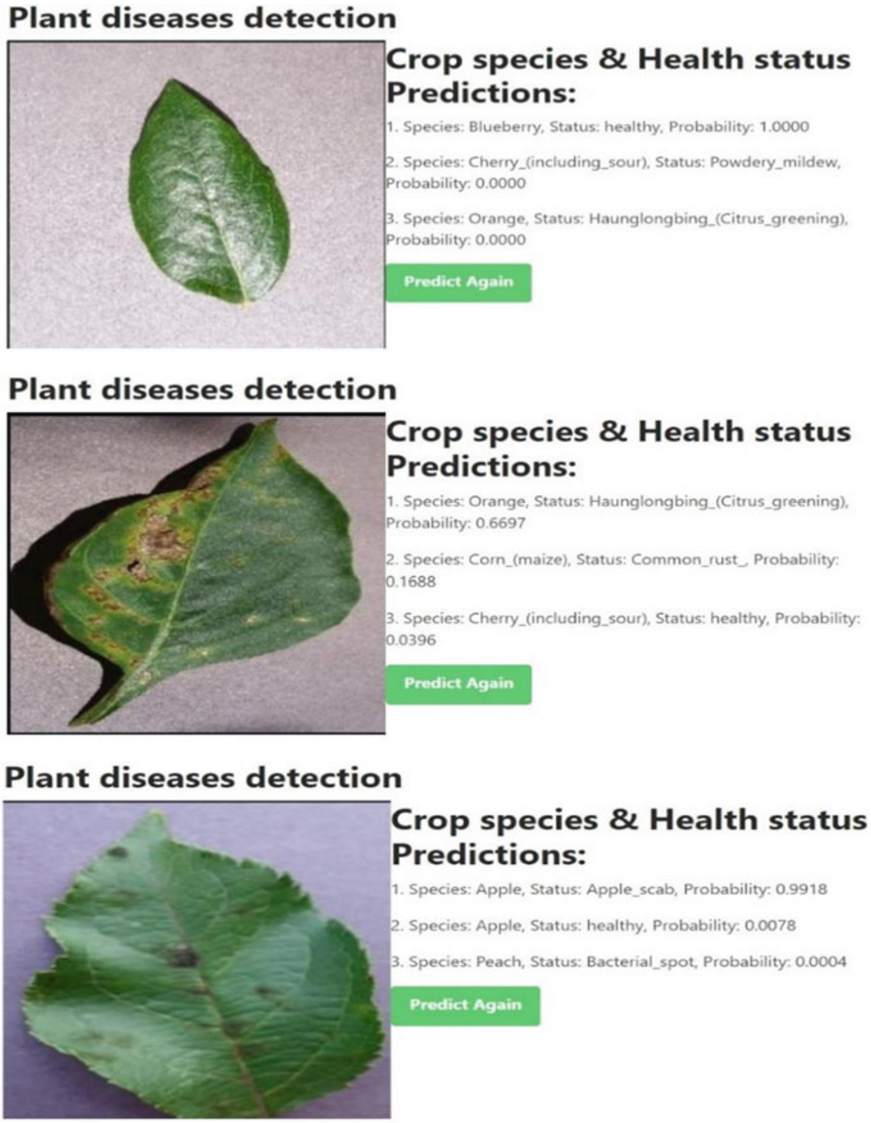
Table 3 depicts the results which were achieved. The report states that Alex Net, VGG-19, VGG-16 need more time to provide precision when compared to ResNet50. The loss experienced from the considered models are relatively higher. In this scenario, the suggested model depicts better precision and performance-rate for detection of images. It is a vital model to recognize crop disease since the structure of leaves show similar anatomy.

A comparative analysis with a former work [18] is depicted in Table 4 as the prior work considered had same number of sub-classes and images. AlexNet and GoogleNet were the two models used previously. The ResNet50 is collated with the outcome of two other models used and received outcome as depicted in Table 4. The image count in the research was similar to the count in previous work. Hence, it becomes easy to come to a better conclusion. It provides excellent accuracy.

As a result of the evaluation conducted, a conclusion is achieved that, ResNet50 performs higher than rest of the models with other pretrained networks.

6 Conclusion

In the research, a webpage was created using Django framework, SQLite database for the backend and HTML, JavaScript and CSS for the frontend. Farmers were able to easily converse on the website by dropping their questions and gaining answers



Plant diseases detection

Crop species & Health status Predictions:

1. Species: Blueberry, Status: healthy, Probability: 1.0000

2. Species: Cherry_(including_sour), Status: Powdery_mildew, Probability: 0.0000

3. Species: Orange, Status: Haunglongbing_(Citrus_greening), Probability: 0.0000

Plant diseases detection

Crop species & Health status Predictions:

1. Species: Orange, Status: Haunglongbing_(Citrus_greening), Probability: 0.6697

2. Species: Corn_(maize), Status: Common_rust_, Probability: 0.1688

3. Species: Cherry_(including_sour), Status: healthy, Probability: 0.0396

Plant diseases detection

Crop species & Health status Predictions:

1. Species: Apple, Status: Apple_scab, Probability: 0.9918

2. Species: Apple, Status: healthy, Probability: 0.0078

3. Species: Peach, Status: Bacterial_spot, Probability: 0.0004

Fig. 12 Some predicted images with the proposed model

Table 2 The model performance of training dataset using CNN Models

Model	Training accuracy	Validation accuracy	Time taken	Loss
Alex Net	0.8496	0.8842	191 s 234 ms/step	0.5500
VGG-19	0.9034	0.9263	293 s 456 ms/step	0.2528
VGG-16	0.9563	1.00	246 s 434 ms/step	0.1523
ResNet50	*0.9998*	*1.00*	*66 s 154 ms/step*	*0.0080*

Table 3 Difference between the accuracies of different CNN models

CNN Models	Percentage (%)	Epochs	Time-taken	Loss generated
Alex Net	95.48	64	227 s 412 ms/step	0.2354
VGG-19	98.27	50	234 s 427 ms/step	0.0646
VGG-16	98.64	48	278 s 489 ms/step	0.0643
ResNet50	*99.77*	*4*	*298 s* *540 ms/step*	*0.0544*

Table 4 Difference Between three models using the dataset

Model considered	Number of images	Sub-classes	Epochs	Total-accuracy (%)
Alex Net [17]	54,306	38	30	97.14
GoogLe Net [17]	54,306	38	30	98.46
ResNet50	*54,309*	*38*	*23*	*99.86*

through the experts signed on the website. A major breakthrough in the project was the plant disease recognition system which help the farmers to detect the disease of 14 various crop species with 38 sub-classes. The Convolutional Neural Network model of ResNet50 architecture was successfully trained and applied to obtain comparative results. The CNN model was able compare 38 sub-classes of healthy as well as diseased leaves. Other Model accuracies were analyzed with suitable examples. It was successfully proved that why ResNet50 architecture is relevant over other models and why this model should be used. The overall accuracy of the test was higher than other models. With the recurring time and change, our agriculture sector is progressing and we aim to solve the problems related to the agriculture to help our fellow farmers. The research will allow the farmers to get the solution to their problems from others those who have experienced the same or from the experts.

References

1. LeCun Y, Bottou L, Bengio Y, Haffner P (1998/99) Object recognition with gradient- based learning
2. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: NeurIPS proceedings
3. Szegedy et al (2014) Going deeper with convolutions. Comput Vis Pattern Recognit
4. Sermanet et al (2013) Overfeat: integrated recognition, localization and detection using convolutional networks
5. Sardogan M, Tuncer A, Ozen Y (2018) Plant leaf disease detection and classification based on CNN with LVQ algorithm. In: 3rd international conference on computer science and engineering (UBMK)

6. Suresh G, Gnanaprakash V, Santhiya R (2019) Performance analysis of different CNN architecture with different optimisers for plant disease classification. In: 5th international conference on advanced computing & communication systems (ICACCS), March 2019
7. Patil JK, Kumar R (2012) Feature extraction of diseased leaf images. J Signal Image Process 3(1):60
8. Pawara P, Okafor E, Surinta O, Schomaker L, Wiering M (2017) Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant detection. In: 6th international conference on pattern recognition applications and methods (ICPRAM 2017), pp 479–486
9. Chaudhary P, Chaudhari AK, Cheeran AN, Godara S (2012) Color transform based approach for disease spot recognition on plant leaf. Int J Comput Sci Telecommun 3(6):65–69
10. Fujita E, Kawasaki Y, Uga H, Kagiwada S, Iyatomi H (2016) Basic investigation on a robust and practical plant diagnostic system. In: 15th IEEE international conference on machine learning and applications (ICMLA 2016), December 2016
11. Khirade SD, Patil AB (2015) Plant disease recognition using image processing. In: 2015 international conference on computing communication control and automation
12. Bashish DA, Braik M, Ahmad SB (2010) A framework for recognition and classification of plant leaf and stem diseases. In: international conference on signal and image processing
13. Sankaran S, Mishra A, Ehsani R, Davis C (2010) A review of advanced techniques for detecting plant diseases. Comput Electron Agric 72(1):1–13
14. Wäldchen J, Mäder P (2018) Plant species identification using computer vision techniques: a systematic literature review. Arch Comput Methods Eng 25(2):507–543
15. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient- based learning applied to document recognition. In: Proceedings of the IEEE, November 1998
16. spMohanty, "plantvillage- dataset" (2018) GitHub repository [Online]. https://github.com/spMohanty/PlantVillage-Dataset. Accessed 13 March 2022
17. Mukti IZ, Biswas D (2019) Transfer learning based plant diseases detection using ResNet50". In: 4th international conference on electrical information and communication technology (EICT), December 2019
18. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease Detection

Capturing Cross-View Dynamics Using Recurrent Neural Networks for Multi-modal Sentiment Analysis

Pranav Chitale, Tanvi Dhope, and Dhananjay Kalbande

Abstract Sentiment analysis through multi-modal approaches has shown the potential to outperform uni-modal approaches. One of the challenges in this domain is to effectively model cross-view dynamics from view-specific dynamics. This paper proposes a model that captures both dynamics, and applies attention over the contributing features from each modality, to predict utterance-level sentiments. In the model, the paper introduces a deep learning pipeline called the Cross-view Recurrent Neural Network Pair to compute cross-view dynamics and integrate them with view-specific dynamics, to obtain contextually rich utterance representations. The proposed model is evaluated on CMU Multi-modal Opinion-level Sentiment Intensity (CMU-MOSI) and CMU Multi-modal Opinion Sentiment and Emotion Intensity (CMU-MOSEI) datasets. The model achieves an accuracy of 81.78% on CMU-MOSI and 80.45% on CMU-MOSEI.

Keywords Multi-modal · Sentiment analysis · Deep learning · Recurrent neural network · Natural language processing

1 Introduction

Sentiment analysis is a widely researched topic in computer science. Traditional machine learning-based approaches involve training a model on data of a single modality such as audio, video, or text. While these approaches may work in general,

P. Chitale (✉)
JPMorgan Chase, Mumbai, India
e-mail: pranavchitale20@gmail.com

T. Dhope
Microsoft, Hyderabad, India
e-mail: tanvidhope@gmail.com

D. Kalbande
Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India
e-mail: drkalbande@spit.ac.in

S. Bhattacharyya et al. (eds.), *Intelligent Systems and Human Machine Collaboration*, Lecture Notes in Electrical Engineering 985,
https://doi.org/10.1007/978-981-19-8477-8_20

they face shortcomings when the stand-alone modality does not provide sufficient information to correctly predict the sentiment. In such cases, processing information from other available modalities helps in obtaining the required context. This approach is known as multi-modal sentiment analysis.

Multi-modal approaches involve training a model on two or more modalities to predict sentiments. The key aspect to a multi-modal model is the coordination between view-specific and cross-view dynamics [1]. Each modality is represented by its characteristic features such as word embedding vectors for text, facial feature descriptors for video, and prosody in speech for audio. By processing a modality individually, meaningful feature representations, referred to as view-specific dynamics, can be extracted. On the other hand, cross-view dynamics refers to the interaction between modalities. These interactions facilitate sharing of information across modalities, resulting in a collective support in favor or against a sentiment.

Some of the recent works explore different techniques in combining the information across modalities. Zadeh et al. [1] process modality sequences individually to obtain embedding vectors for each of them, and then fuse the vectors in uni-modal, bi-modal, and tri-modal formats. Ghosal et al. [2] propose a pairwise attention mechanism for combining utterance-level modality features. Zadeh et al. [3] present a different approach by taking the view-specific features at each time-step, computing cross-view dynamics from them through an attention mechanism, and using these dynamics to process the next time-step. Similarly, [4] uses LSTMs to process each modality, tracks changes in their outputs with respect to the previous time-step, and uses a gating mechanism to store the cross-view interactions over time. The current approaches treat cross-view dynamics to be common across all modalities by combining them into a single feature representation. This opens up areas to explore techniques allowing each modality to have cross-view dynamics specific to itself.

This paper proposes a model employing a novel deep learning pipeline called the Cross-view Recurrent Neural Network (RNN) Pair for modeling cross-view dynamics, and integrating them with view-specific dynamics. In this approach, two pathways are provided, namely, primary and secondary RNNs, for each modality to process the view-specific and cross-view dynamics, respectively. A primary RNN is a recurrent unit that contains an additional hidden state vector to store cross-view information. The output of a primary RNN at each time-step is computed by including this cross-view information, along with other variables of the RNN. The output vectors thus obtained are taken in pairs of modalities, and then processed by secondary RNNs to obtain the third modality's cross-view dynamics for the next time-step. The secondary RNN processes and updates cross-view dynamics for each time-step, maintaining information that is relevant. In this way, the paper addresses the following two challenges—(1) computation of cross-view dynamics specific to each modality and (2) integrating cross-view dynamics with view-specific dynamics.

The remaining sections of the paper are organized as follows: Sect. 2 gives an overview of the previous work on multi-modal sentiment analysis. Section 3 discusses the proposed solution and overall model architecture in detail. Section 4 provides dataset details, model parameters, experimental results, and qualitative analysis. Finally, Sect. 5 summarizes and concludes this paper.

2 Literature Review

Multi-modal approaches for sentiment analysis or emotion recognition broadly include the following two tasks:

1. Feature extraction for each modality.
2. Effective processing of multi-modal features.

There are many popular tools for feature extraction such as BERT or GloVe embeddings for textual utterances, FACET [5] for visual features, and COVAREP [6] for extracting acoustic features like Mel-Frequency Cepstral Coefficients (MFCCs), pitch, etc. The feature representations obtained from such tools can then be used for training and prediction.

Previous research on sentiment analysis and emotion recognition can be further divided into two parts:

1. Uni-modal models—where a single mode (either audio, video, or text) is used. Since only one mode is involved, the accuracy of these models is fully dependent on the quality of features in that modality.
2. Multi-modal models—where more than one modes are used in coherence. Since there are more data points in a multi-modal space, the accuracy of these models is usually higher than their uni-modal counterpart.

There are various methods used for sentiment analysis and emotion recognition, both uni-modal and multi-modal. Cai et al. [7] use Bi-LSTM for high-level contextual feature extraction, and CNN for feature extraction from video, obtaining an accuracy of 70.4%. The accuracy of the model may possibly be improved by taking into consideration textual features as well. Another work along similar lines is [8], which uses Factorized Bilinear Pooling (FBP) for feature fusion. Febriansyah et al. [9] use the Toronto Emotional Speech Set (TESS) and Extreme Learning Machine for feature extraction like MFCC, pitch, and intensity. It was able to classify all the data points correctly over the small dataset considered.

There are a few noteworthy models which use complex inter-modality and intra-modality interactions for better accuracy. One such example is MISA [10], which computes two representations—one specific to modality and other independent of it. Fusion of modalities is accomplished through transformer network. MISA has been evaluated for multi-modal sentiment analysis and humor detection. The authors of [11] extract high-level features from the raw modalities and process them in two levels using different types of LSTMs. Fusion of modalities is performed by concatenation of the outputs from the first LSTM level. [12] factorizes multi-modal representations into multi-modal discriminative factors and modality-specific generative factors. The work is evaluated over six datasets including CMU-MOSI, the binary accuracy for the same being 78.1% for sentiment analysis task. The authors of [13] use low-rank tensors to improve efficiency, enabling their model to scale linearly with number of modalities. The model was able to achieve a binary accuracy of 76.4% on CMU-MOSI dataset. References [14, 15] propose using a system of Gated Recurrent Units

(GRU) to model a conversation, process the state of the speakers, and track the context. Both approaches are trained to classify emotions, while the authors of [14] also apply their model for sentiment analysis.

In the previous research carried out, modeling of proper cross-view dynamics has been the challenging factor for many models. Various methods have been tried and tested with varying degrees of success. In this paper, Cross-view RNN Pair is introduced to address this issue and a model architecture around it is proposed.

3 Proposed Methodology

In the proposed model, the aim is to capture the contextual features from the input utterance sequences to predict the sentiments at utterance level. The input sequences contain multi-modal (i.e., acoustic, visual, and text) time distributed information obtained from a video of a speaker. The overall proposed model architecture (Fig. 1) comprises three parts, each contributing toward learning a particular aspect of multi-modal features.

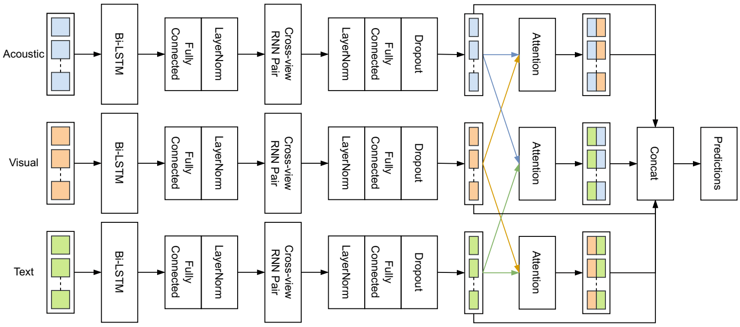


Fig. 1 Overall architecture of proposed model

First, the input sequence of utterances of each modality is processed by a Bi-directional Long Short-Term Memory (Bi-LSTM), a fully connected layer, followed by layer normalization [16]. Output of this part is the view-specific representation of the utterance sequences. The second part of the proposed model aims to learn cross-view dynamics by leveraging the view-specific dynamics from the first step. For this, the paper introduces Cross-view Recurrent Neural Network (RNN) Pair (Fig. 2), a pipeline for processing multi-modal input sequences with the means for sharing information across modalities. In this pipeline, for each modality, a pair of primary and secondary RNNs are used to process view-specific and cross-view dynamics, respectively. A primary RNN is any recurrent unit with an additional hidden state vector to store cross-view information. In this paper, Long Short-Term Hybrid Memory (LSTHM) [3] cell is used for the primary RNN. LSTHM is an extension of LSTM, with an additional memory component to store cross-view dynamics. It takes two inputs, where the model provides the view-specific dynamics, and a component called cross-view context that accounts for cross-view dynamics. The cross-view context for a given modality and time-step is computed from LSTHM outputs of the previous time-step, by combining and processing them using secondary RNNs. For the secondary RNN, LSTM [17] is used in this paper. The approach starts by pairing up the LSTHM outputs of the previous time-step modality-wise (i.e., acoustic-visual, visual-text, acoustic-text), and performs an element-wise matrix sum operation on each pair, to obtain view-complement for the third modality (e.g., acoustic view-complement is derived from visual-text pair). For a given modality, its view-complement contains information from other modalities at a given time-step. The view-complements are then passed to the secondary RNNs of the corresponding modality. The outputs of secondary RNNs are used as the cross-view contexts for the LSTHM in the next time-step. Finally, the output of each modality's primary RNN is collected at every time-step, and then passed through layer normalization, a fully connected layer, and a dropout layer to obtain the final utterance representations. The last part of the proposed model takes the final utterance representations and applies Multi-Modal Multi-Utterance-Bi-Modal Attention (MMMU-BA) [2] over them. The MMMU-BA framework puts focus on contributing features across modalities and utterances. The attended utterance sequences from the MMMU-BA framework are concatenated with primary RNN's output sequences at the utterance level. This concatenated matrix is passed through a fully connected layer, whose outputs are used for predictions. The softmax activation function is applied on these outputs to obtain class probabilities in classification tasks, whereas the fully connected layer's output is directly used as predicted value in regression tasks.

The following subsections provide more details about the flow of the proposed model.

3.1 View-Specific Utterance Representation

The input utterance sequences are denoted by $X^m = \{x_1^m, x_2^m, ..., x_T^m : m \in M, x_t^m \in \mathbb{R}^{d_{in}^m}\}$, where T is the total number of time-steps/utterances, set M includes the three modalities—acoustic (a), visual (v), and text (t). The dimensionality of an input utterance x_t^m of modality m is d_{in}^m. Each of the M modality sequences—X^a, X^v, and X^t—is fed to a Bi-LSTM, and the outputs of all time-steps are collected. The utterance-level output sequences are then passed through a fully connected layer with d_{FC1} hidden units, followed by layer normalization, resulting in view-specific utterance representation $U^m = \{u_1^m, u_2^m, ..., u_T^m : m \in M, u_t^m \in \mathbb{R}^{d_{FC1}}\}$. The representations thus obtained capture the view-specific dynamics.

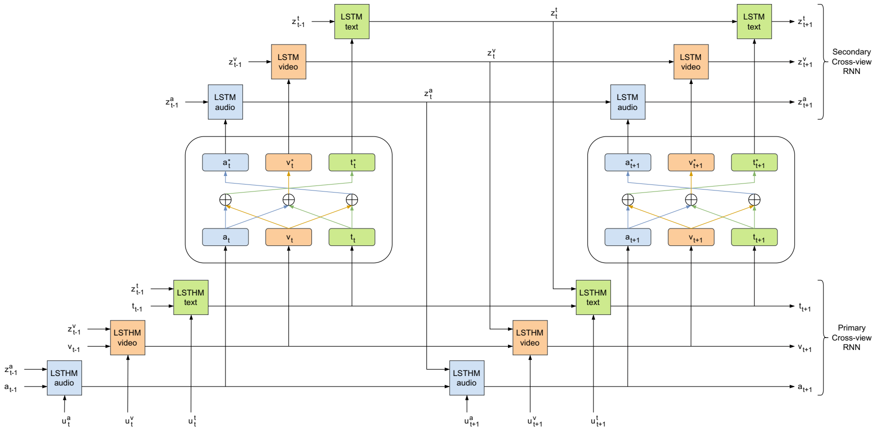


Fig. 2 Cross-View RNN pair

3.2 *Cross-View RNN Pair*

For the next part, the paper uses a system of LSTHMs and LSTMs (Fig. 2) to model the cross-view dynamics from the view-specific dynamics. LSTHM cells are used for the primary cross-view RNNs, to extract the view-specific utterances and to integrate with cross-view dynamics. As inputs to LSTHM of modality m at time-step t, the view-specific utterance representation u_t^m and the cross-view context of the previous time-step $z_{t-1}^m \in \mathbb{R}^{d_{CR2}}$ are provided, to obtain outputs a_t, v_t, and t_t as follows:

$$a_t = \mathrm{LSTHM}(u_t^a, a_{t-1}, z_{t-1}^a) \tag{1.1}$$

$$v_t = \mathrm{LSTHM}(u_t^v, v_{t-1}, z_{t-1}^v) \tag{1.2}$$

$$t_t = \mathrm{LSTHM}(u_t^t, t_{t-1}, z_{t-1}^t). \tag{1.3}$$

The view-complements a_t^*, v_t^*, and t_t^* are then computed by taking element-wise matrix sum (denoted by $\oplus$) on modality pairs (v_t, t_t), (a_t, t_t), and (a_t, v_t), respectively. A view-complement represents for a modality, the context provided by other modalities. It is defined as follows:

$$a_t^* = v_t \oplus t_t \tag{2.1}$$

$$v_t^* = a_t \oplus t_t \tag{2.2}$$

$$t_t^* = a_t \oplus v_t. \tag{2.3}$$

The view-complements thus obtained are passed as input to a secondary cross-view RNN. For this, an LSTM of output dimensionality d_{CR2} is used, to model the cross-view context sequence $Z^m = \{z_1^m, z_2^m, ..., z_T^m : m \in M, z_t^m \in \mathbb{R}^{d_{CR2}}\}$. Inputs to the secondary RNNs at time-step t are a_t^*, v_t^*, and t_t^*.

$$z_t^a = \mathrm{LSTM}(a_t^*, z_{t-1}^a) \tag{3.1}$$

$$z_t^v = \mathrm{LSTM}(v_t^*, z_{t-1}^v) \tag{3.2}$$

$$z_t^t = \mathrm{LSTM}(t_t^*, z_{t-1}^t). \tag{3.3}$$

z_t^a, z_t^v, and z_t^t are then taken as the cross-view context in primary RNNs for the next time-step, and the process continues for all T time-steps.

Finally, as outputs of the Cross-view RNN Pair, the obtained matrices contain acoustic, visual, and text utterance representations $[a_1, a_2, ..., a_T]$, $[v_1, v_2, ..., v_T]$, and $[t_1, t_2, ..., t_T]$, respectively, each of dimensionality $\mathbb{R}^{T \times d_{CR1}}$. These three matrices are then passed through layer normalization, a fully connected layer with d_{FC2} hidden units, and a dropout layer. The resulting matrices H^a, H^v, and $H^t \in \mathbb{R}^{T \times d_{FC2}}$ are the final utterance representations.

3.3 Attention Layer

In order to attend to contributing features within the final utterance representations, the model uses an attention mechanism proposed by [2]. The approach computes attention over utterance sequences in a pairwise manner, for which the matrix pairs (H_a, H_v), (H_v, H_t), and (H_a, H_t) are used. The pairwise attention for (H_a, H_v) is obtained as follows:

$$P_1 = H^a \cdot (H^v)^T \quad \& \quad P_2 = H^v \cdot (H^a)^T \tag{4.1}$$

$$Q_1(i, j) = \frac{\exp(P_1(i, j))}{\sum_{k=1}^{T} \exp(P_1(i, k))} \quad \text{for} \quad i, j = 1, ..., T \tag{4.2}$$

$$Q_2(i, j) = \frac{\exp(P_2(i, j))}{\sum_{k=1}^{T} \exp(P_2(i, k))} \quad \text{for} \quad i, j = 1, ..., T \tag{4.3}$$

$$R_1 = Q_1 \cdot H^v \quad \& \quad R_2 = Q_2 \cdot H^a \tag{4.4}$$

$$A_1 = R_1 \odot H^a \quad \& \quad A_2 = R_2 \odot H^v \tag{4.5}$$

$$A^{av} = \mathrm{concat}[A_1, A_2] \tag{4.6}$$

where $P_1, P_2 \in \mathbb{R}^{T\times T}$; $Q_1, Q_2 \in \mathbb{R}^{T\times T}$; $R_1, R_2 \in \mathbb{R}^{T\times d_{FC2}}$; $A_1, A_2 \in \mathbb{R}^{T\times d_{FC2}}$; $A^{av} \in \mathbb{R}^{T\times 2d_{FC2}}$; and $\odot$ denotes element-wise matrix product. Similarly, attention is applied over the pairs (H_v, H_t), (H_a, H_t) to obtain pairwise attentions A^{vt}, $A^{at} \in \mathbb{R}^{T\times 2d_{FC2}}$, respectively. A detailed explanation for the working of attention mechanism and the equations involved is provided by [2].

3.4 Predictions

Predictions are made at the utterance level, for each utterance in the multi-modal input sequences. The pairwise attentions A^{av}, A^{vt}, and A^{at} are taken along with sequences H^a, H^v, and H^t to compute matrix C, comprising concatenated vectors for T utterances, as follows:

$$C = \text{concat}[A^{av}, A^{vt}, A^{at}, H^a, H^v, H^t], \tag{5}$$

where $C \in \mathbb{R}^{T\times 9d_{FC2}}$, and $c_t \in C$ denotes the concatenated vector for the utterance at time-step t.

The model's predictions are computed differently, depending on whether the task is classification or regression.

3.4.1 Classification

In classification tasks, such as emotion recognition or sentiment classification, taking N as the number of classes, c_t is passed through a fully connected layer with N output units, followed by the softmax activation function to obtain class probabilities $y_t^c \in \mathbb{R}^N$. y_t^c represents the output for an utterance at time-step t.

$$y_t^c = \text{softmax}(\text{FC}(c_t)). \tag{6.1}$$

3.4.2 Regression

In regression tasks, a fully connected layer with a single output unit is used to obtain the sentiment intensity prediction $y_t^r \in \mathbb{R}^1$, for an utterance at time-step t.

$$y_t^r = \text{FC}(c_t). \tag{6.2}$$

4 Results and Discussion

4.1 Datasets and Features

For the purpose of evaluating the proposed model, the following two benchmark datasets for multi-modal sentiment analysis are used—CMU Multi-modal Opinion-level Sentiment Intensity (CMU-MOSI) [18] and CMU Multi-modal Opinion Sentiment and Emotion Intensity (CMU-MOSEI) [19].

4.1.1 CMU-MOSI

The CMU-MOSI dataset comprises 93 videos. Each video is segmented into a sequence of utterances. There are 1151, 296, and 752 utterances in train, validation, and test set, respectively, and 2199 utterances collectively. A sentiment intensity score in the continuous inclusive range of -3 (strong negative) to +3 (strong positive) is provided for each utterance. For CMU-MOSI, the utterance-level features provided in [11] are used, where the dimensions of utterance-level inputs are 73, 100, and 100 for acoustic, visual, and text, respectively.

4.1.2 CMU-MOSEI

The CMU-MOSEI dataset has 3229 videos, with 22676 utterances in total. The train, validation, and test set splits contain 16216, 1835, and 4625 utterances, respectively. CMU-MOSEI provides for the sentiment intensity scores in the same range as CMU-MOSI. In addition, it includes labels to represent the presence of the following six emotions—happiness, sadness, anger, surprise, disgust, and fear. For CMU-MOSEI, the features provided in [14] are used, where the dimensions of utterance-level inputs are 384, 35, and 300 for acoustic, visual, and text, respectively.

4.2 Methodology Setup

In the proposed model, number of hidden units in Bi-LSTMs are set as 128, 128, 128 (MOSI) and 384, 35, 300 (MOSEI) for acoustic, visual, and text sequences. The size of fully connected layer for view-specific utterance representation is set as d_{FC1} = 128 (MOSI) and 256 (MOSEI), and for the final utterance representation d_{FC2} = 64 (MOSI) and 100 (MOSEI). In Cross-view RNN Pair, the primary cross-view RNN's output size is set as d_{CR1} = 128 (MOSI) and 256 (MOSEI), and the secondary cross-view RNN's output, i.e., cross-view context size is set as d_{CR2} = 128 (MOSI) and 128 (MOSEI).

The model is trained for 50 epochs using Adam optimizer [20] with learning rate $\alpha = 0.0001$ for both datasets. The batch size is set as 32 for MOSI and 128 for MOSEI. The dropout used for MOSI and MOSEI sentiment classification is 0.5, and 0.3 for MOSEI sentiment score prediction.

4.3 Experimental Outcome

The proposed model is evaluated for binary sentiment classification on CMU-MOSI and CMU-MOSEI using accuracy (A^2) and F1 score. To obtain the two classes, the sentiment intensity score labels ≥ 0 are taken as the positive sentiment class, while labels < 0 represent negative sentiment. The model is also evaluated for regression on sentiment intensity score labels on CMU-MOSEI using Mean Absolute Error (MAE). Results of the proposed model are compared with other models in their tri-modal setup.

Table 1 Comparison of the proposed model's results on CMU-MOSI with other models (following [14])

Model	A^2	F1
bc-LSTM	80.30	–
MMMU-BA	**82.31**	–
DialogueRNN	79.80	79.48
Multilogue-net	81.19	80.10
Proposed model	81.78	**81.58**

Table 1 presents the comparison of the proposed model's performance on CMU-MOSI with the models bc-LSTM [11], MMMU-BA [2], DialogueRNN [15], and Multilogue-net [14]. A^2 is reported for all five models. The proposed model obtained a binary accuracy of 81.78%, an improvement of 0.59% over Multilogue-net, but 1.12% lower than MMMU-BA, which achieved the highest accuracy at 82.31%. In terms of F1 score, the proposed model obtained 81.58%, an overall improvement as compared to 79.48% of DialogueRNN and 80.10% of Multilogue-net. F1 score was not reported for bc-LSTM and MMMU-BA in [14] for CMU-MOSI.

Table 2 shows the performance of the proposed model on CMU-MOSEI in comparison with Graph-MFN [4], MMMU-BA [2], DialogueRNN [15], and Multilogue-net [14]. Binary accuracy A^2 is reported for all five models. The proposed model achieves 80.45%, an improvement compared to other models, but falls short by 1.65% with respect to Multilogue-net, which achieves the highest accuracy at 82.10%. F1 score and MAE were reported for Graph-MFN, DialogueRNN, and Multilogue-net. The proposed model achieved an overall improvement in these metrics with an F1 score of 81.20% and 0.58 MAE.

Table 2 Comparison of the proposed model's results on CMU-MOSEI with other models (following [14])

Model	A^2	F1	MAE
Graph-MFN	76.90	77.00	0.71
MMMU-BA	79.80	–	–
DialogueRNN	79.98	79.82	0.69
Multilogue-net	**82.10**	80.01	0.59
Proposed model	80.45	**80.20**	**0.58**

4.4 Result Analysis

This section analyzes the performance of the proposed model on sentiment classification task. On CMU-MOSI, the proposed model obtains a precision and recall of 83.40% and 88.22% for positive sentiment and 78.68% and 71.23% for negative sentiment. For CMU-MOSEI, on the other hand, the precision and recall are observed to be 87.42% and 88.71% for positive sentiment and 47.61% and 44.57% for negative sentiment. The proposed model struggles on CMU-MOSEI sentiment classification task, as opposed to a fair performance on CMU-MOSI. Class imbalance is theorized to be the impeding factor affecting the performance of the proposed model. This is backed by the fact that CMU-MOSI has well-balanced binary sentiment labels, while CMU-MOSEI is skewed toward positive sentiments; the ratio of positive to negative sentiments is approximately 2.3:1.

5 Conclusion

In this paper, a model with recurrent neural network-based architecture for multimodal sentiment analysis was proposed. The model learns from utterance-level acoustic, visual, and text sequences. Through the proposed Cross-view RNN Pair pipeline, the model effectively formulates cross-view dynamics and integrates them with view-specific dynamics to obtain contextually rich utterance representations. Evaluation of the proposed model on benchmark datasets, such as CMU-MOSI and CMU-MOSEI, has shown that it performs fairly well compared to recent works.

The future scope of work will be to explore mechanisms that detect the presence of multiple emotions, while keeping the model relatively more robust to class imbalance.

References

1. Zadeh A, Chen M, Poria S, Cambria E, Morency LP (2017) Tensor fusion network for multimodal sentiment analysis. In: Proceedings of the 2017 conference on empirical methods in natural language processing. Association for computational linguistics, copenhagen, Denmark, pp 1103–1114. https://aclanthology.org/D17-1115
2. Ghosal D, Akhtar MS, Chauhan D, Poria S, Ekbal A, Bhattacharyya P (2018) Contextual intermodal attention for multi-modal sentiment analysis. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for computational linguistics, Brussels, Belgium (Oct-Nov 2018), pp 3454–3466. https://aclanthology.org/D18-1382
3. Zadeh A, Liang PP, Poria S, Vij P, Cambria E, Morency LP (2018) Multi-attention recurrent network for human communication comprehension. In: Proceedings of the Thirty-second aaai conference on artificial intelligence and thirtieth innovative applications of artificial intelligence conference and eighth AAAI symposium on educational advances in artificial intelligence. AAAI'18/IAAI'18/EAAI'18, AAAI Press
4. Zadeh A, Liang PP, Mazumder N, Poria S, Cambria E, Morency LP (2018) Memory fusion network for multi-view sequential learning. https://arxiv.org/abs/1802.00927
5. iMotions: (2017). https://goo.gl/1rh1JN, facial Expression Analysis
6. Degottex G, Kane J, Drugman T, Raitio T, Scherer S (2014) Covarep - a collaborative voice analysis repository for speech technologies. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 960–964
7. Cai L, Dong J, Wei M (2020) Multi-modal emotion recognition from speech and facial expression based on deep learning. In: 2020 Chinese automation congress (CAC), pp 5726–5729
8. Zhou H, Du J, Zhang Y, Wang Q, Liu QF, Lee CH (2021) Information fusion in attention networks using adaptive and multi-level factorized bilinear pooling for audio-visual emotion recognition. IEEE/ACM Trans Audio Speech Lang Process 29:2617–2629
9. Ainurrochman, Febriansyah II, Yuhana UL (2021) Ser: speech emotion recognition application based on extreme learning machine. In: 2021 13th International Conference on Information & Communication Technology and System (ICTS), pp 179–183 (2021)
10. Hazarika D, Zimmermann R, Poria S (2020) Misa: modality-invariant and -specific representations for multimodal sentiment analysis (2020). https://arxiv.org/abs/2005.03545
11. Poria S, Cambria E, Hazarika D, Majumder N, Zadeh A, Morency LP (2017) Context-dependent sentiment analysis in user-generated videos. In: Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers). Association for computational linguistics, vancouver, Canada, pp 873–883. https://aclanthology.org/P17-1081
12. Tsai YHH, Liang PP, Zadeh A, Morency LP, Salakhutdinov R (2018) Learning factorized multimodal representations. https://arxiv.org/abs/1806.06176
13. Liu Z, Shen Y, Lakshminarasimhan VB, Liang PP, Bagher Zadeh A, Morency LP (2018) Efficient low-rank multimodal fusion with modality-specific factors. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers). Association for computational linguistics, Melbourne, Australia, pp 2247–2256.https://aclanthology.org/P18-1209
14. Shenoy A, Sardana A (2020) Multilogue-net: a context-aware RNN for multi-modal emotion detection and sentiment analysis in conversation. In: Second grand-challenge and workshop on multimodal language (Challenge-HML), pp 19–28. Association for Computational Linguistics, Seattle, USA. https://aclanthology.org/2020.challengehml-1.3
15. Majumder N, Poria S, Hazarika D, Mihalcea R, Gelbukh A, Cambria E (2018) Dialoguernn: an attentive rnn for emotion detection in conversations. https://arxiv.org/abs/1811.00405
16. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. https://arxiv.org/abs/1607.06450
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Zadeh A, Zellers R, Pincus E, Morency LP (2016) Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. https://arxiv.org/abs/1606.06259

19. Bagher Zadeh A, Liang PP, Poria S, Cambria E, Morency LP (2018) Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers). Association for computational linguistics, Melbourne, Australia, pp 2236–2246. https://aclanthology.org/P18-1208
20. Kingma DP, Ba J (2017) Adam: a method for stochastic optimization