Advances in Intelligent Systems and Computing 799

Nishchal K. Verma · A. K. Ghosh
Editors

Computational Intelligence: Theories, Applications and Future Directions—Volume II

ICCI-2017

Springer

Advances in Intelligent Systems and Computing

Volume 799

Series editor

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

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment,Web intelligence and multimedia.

The publications within "Advances in Intelligent Systems and Computing" are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

Chairman

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

Members

Rafael Bello Perez, Faculty of Mathematics, Physics and Computing, Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

Hani Hagras, School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK

László T. Kóczy, Faculty of Engineering Sciences, Department of Information Technology, Széchenyi István University, Győr, Hungary

Vladik Kreinovich, Department of Computer Science, University of Texas at El Paso, El Paso, TX, USA

Chin-Teng Lin, Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan

Jie Lu, Faculty of Engineering and Information, University of Technology, Sydney, NSW, Australia

Patricia Melin, Graduate Program of Computer Science, Tijuana Institute of Technology, Tijuana, Mexico

Nadia Nedjah,Department of Electronics Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil

Ngoc Thanh Nguyen, Wrocław University of Science and Technology, Wrocław, Poland

Jun Wang, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong

More information about this series at http://www.springer.com/series/11156

Nishchal K. Verma · A. K. Ghosh
Editors

Computational Intelligence: Theories, Applications and Future Directions—Volume II

ICCI-2017

Springer

Editors
Nishchal K. Verma
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh, India

A. K. Ghosh
Department of Aerospace Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh, India

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-981-13-1134-5 ISBN 978-981-13-1135-2 (eBook)
https://doi.org/10.1007/978-981-13-1135-2

Library of Congress Control Number: 2018946601

© Springer Nature Singapore Pte Ltd. 2019
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

Computational intelligence (CI) focuses on the biologically and linguistically inspired computing tools and techniques. CI technologies have been playing and will continue to play important roles in a wide variety of application areas. Although there have been many success stories including techniques like deep learning, a lot more is left to be done, particularly if one wants computing tools to really mimic the brain either at a lower (neuronal) level or at a higher level. The scientific exchanges in various collocated events are expected to play important roles in advancing CI theories, applications and future directions.

The modern world systems and processes are growing exponentially in size with complexity; therefore, their monitoring and control is a challenging task. CI provides a set of tools, namely, but not limited to neural networks, evolutionary algorithms, fuzzy systems and hybrid intelligent systems, that can understand the principles which make the intelligent behaviour possible in these systems. Through this volume, researchers, scientists, academicians, scholars and practitioners in this area will get an exposure to recent innovations and initiatives using CI-based approaches for addressing the complex real-world problems. The deliberations should stimulate new thoughts and ideas and pave the way for fruitful collaborations, explorations and interdisciplinary applications in CI.

Kanpur, India

Nishchal K. Verma
A. K. Ghosh

Contents

Part I Intelligent Informatics

Fog Computing-Based Autonomic Security Approach to Internet of Things Applications 3
S. Prabavathy, K. Sundarakantham, S. Mercy Shalinie and K. Narasimha Mallikarjunan

Self-adaptive Frequent Pattern Growth-Based Dynamic Fuzzy Particle Swarm Optimization for Web Document Clustering 15
Raja Varma Pamba, Elizabeth Sherly and Kiran Mohan

A Feature-Reduced Discretized Random Forest Model for Oral Bioavailability Data Classification 27
Priyanka Shit and Haider Banka

TelNEClus: Telugu Named Entity Clustering Using Semantic Similarity 39
SaiKiranmai Gorla, Aditya Chandrashekhar, N. L. Bhanu Murthy and Aruna Malapati

Convolutional Neural Network with Stacked Autoencoder for Kernel Initialization 53
Vikas Singh, Anirudh Swaminathan and Nishchal K. Verma

Behavioural Analysis of Recent Ransomwares and Prediction of Future Attacks by Polymorphic and Metamorphic Ransomware 65
Navneet Kaur Popli and Anup Girdhar

Improved EMD Local Energy with SVM for Fault Diagnosis in Air Compressor 81
Seetaram Maurya, Vikas Singh, Narendra Kumar Dhar and Nishchal K. Verma

Simultaneous Structure and Parameter Learning of Convolutional Neural Network 93
Soniya, Sandeep Paul and Lotika Singh

Part II Intelligent Social Informatics

Hybrid Approach for Predicting and Recommending Links in Social Networks 107
Shashi Prakash Tripathi, Rahul Kumar Yadav, Abhay Kumar Rai and Rajiv Ranjan Tewari

Online Partitioning of Large Graphs for Improving Scalability in Recommender Systems 121
Vinit Koshti, NVS Abhilash, Karanjit S. Gill, Nikhil Nair, Mahima B. Christian and Prosenjit Gupta

Fuzzy Rough Set-Based Feature Selection with Improved Seed Population in PSO and IDS 137
Tarun Maini, Abhishek Kumar, Rakesh Kumar Misra and Devender Singh

Domain-Independent Sentiment Analysis in Malayalam 151
V. Kasthoori, B. Soniya and V. Jayan

Path Planning for Multiple Mobile Robots by Priority Assignment 161
Ram Kishan Dewangan, Anupam Shukla and W. Wilfred Godfrey

Part III Neural Networks

Modified Back Propagation Neural Network for Facial Expression Classification Using Principal Component Analysis and Ridgelet Transform 175
Ramachandran Vedantham, Lavanya Settipalli and Edara Sreenivasa Reddy

Classification of Hand Movement Stages for Brain–Computer Interface Using Convolutional Neural Network 189
Kriti Singhal, Evi Agarwal, Arpit Yadav and Anuraj Singh

Neural Net Optimization by Weight-Entropy Monitoring 201
Seba Susan, Rohit Ranjan, Udyant Taluja, Shivang Rai and Pranav Agarwal

Multimodal Data Fusion and Prediction of Emotional Dimensions Using Deep Neural Network 215
Dhruv Bhandari, Sandeep Paul and Apurva Narayan

Autoencoding Convolutional Representations for Real-Time Eye-Gaze Detection 229
Tharun Kumar Reddy, Vinay Gupta and Laxmidhar Behera

Hausdorff Distance-Based Binary Search Tree Multiclass Decomposition Algorithm 239
Rahul Kumar Sevakula and Nishchal K. Verma

Part IV Neuro-Fuzzy Systems

Investigation of RBF Kernelized ANFIS for Fault Diagnosis in Wireless Sensor Networks 253
Rakesh Ranjan Swain, Tirtharaj Dash and Pabitra Mohan Khilar

On Adaptive Control for AGV with Vision Sensor as an Unknown Nonlinear System 265
Pankaj Kumar Mishra, Pooja Agrawal, Narendra Kumar Dhar and Nishchal K. Verma

A Modified Extreme Learning ANFIS for Higher Dimensional Regression Problems 279
C. P. Pramod, Mahendra Singh Tomar and G. N. Pillai

Part V Object Identification and Scene Analysis

Analysis of Transfer and Residual Learning for Detecting Plant Diseases Using Images of Leaves 295
Ines Khandelwal and Sundaresan Raman

Batch-Mode Active Learning-Based Superpixel Library Generation for Very High-Resolution Aerial Image Classification 307
Rajeswari Balasubramaniam, Srivalsan Namboodiri, Gorthi. R. K. Sai Subrahmanyam and Rama Rao Nidamanuri

An Efficient Algorithm for Image Haze Removal in Outdoor Environment 319
Himanshu Dua, Teena Sharma, Pooja Agrawal and Nishchal K. Verma

Multi-faced Object Recognition in an Image for Inventory Counting 333
Teena Sharma, Shreedharkumar D. Rajurkar, Nikhil Molangur, Nishchal K. Verma and Al Salour

Detection of Dust Deposition Using Convolutional Neural Network for Heritage Images 347
Teena Sharma, Pooja Agrawal and Nishchal K. Verma

Railway Corridor Monitoring Using Deep Drone Vision 361
Sushanth Ikshwaku, Amrutha Srinivasan, Ashley Varghese and Jayavardhana Gubbi

SURF–MSER Based 3D Mapping Using RGB-D Camera on Automated Vehicle 373
Nishchal K. Verma, Aquib Mustafa, Narendra Kumar Dhar and Vibhav Sarraf

Rotation-Invariant Descriptor for Disparate Images Using Line Segments 387
Piyush Sahoo, Teena Sharma, Pooja Agrawal and Nishchal K. Verma

Object Detection and Recognition Using Small Labeled Datasets 407
Akhilesh Raj, Kanishk Gandhi, Bhanu Teja Nalla and Nishchal K. Verma

Part VI Optimization Algorithms

Multi-objective OPF Problem Analysis with Practical Constraints in the Presence of FACTS Devices Using NSHCSA 423
M. Balasubbareddy

A Unified TOPSIS Approach to MADM Problems in Interval-Valued Intuitionistic Fuzzy Environment 435
Samir Kumar and Animesh Biswas

Computational Intelligence for Localization of Mobile Wireless Sensor Networks 449
Vaishali R. Kulkarni and Veena Desai

Reasoning for Uncertainty and Rough Set-Based Approach for an Efficient Biometric Identification: An Application Scenario 465
Ajeet Singh, Vikas Tiwari, Priyanka Garg and Appala Naidu Tentu

Butterfly Constrained Optimizer for Constrained Optimization Problems 477
Abhishek Kumar, Tarun Maini, Rakesh Kumar Misra and Devender Singh

Multimodal Medical Image Fusion Based on Fuzzy Sets with Orthogonal Teaching–Learning-Based Optimization 487
T. Tirupal, B. Chandra Mohan and S. Srinivas Kumar

Inspection–Repair-Based Availability Optimization of Distribution System Using Bare Bones Particle Swarm Optimization 501
Aditya Tiwary

Butterfly Optimizer for Placement and Sizing of Distributed Generation for Feeder Phase Balancing 519
Sujeet Mishra, Abhishek Kumar, Devender Singh and Rakesh Kumar Misra

Differential Evolution-Based Matched Wavelet for Differential Protection of Transformer 531
Arpita Roy, Devender Singh and Rakesh K. Misra

Part VII Pattern Recognition and Medical Imaging

Dynamic Texture Recognition from Multi-offset Temporal Intensity Co-occurrence Matrices with Local Pattern Matching 545
Seba Susan, Minni Mittal, Srishti Bansal and Prachi Agrawal

Multimodal Biometric Authentication System Using Hand Shape, Palm Print, and Hand Geometry 557
Gaurav Jaswal, Amit Kaul and Ravinder Nath

Digital Image Restoration of Historical Devanagari Manuscripts 571
Nidhi Dubey

Emotion Classification from Facial Expressions Using Cascaded Regression Trees and SVM 585
Manjot Singh Bilkhu, Samarth Gupta and Vinay K. Srivastava

Multi-layer Classification Approach for Online Handwritten Gujarati Character Recognition 595
Vishal A. Naik and Apurva A. Desai

A Local Information-Based Fuzzy C-Means for Brain MRI Segmentation 607
Anu Mangla and Chandan Singh

Analysis and Classification of Discriminative Region in Cognitive Functional MRI Data 621
J. Siva Ramakrishna and Hariharan Ramasangu

Face Image Super-Resolution Using Differential Evolutionary Algorithm 635
Shyam Singh Rajput, K. V. Arya and Vijay Kumar Bohat

Noise-Robust Low-Resolution Face Recognition Using SIFT Features 645
K. V. Arya, Shyam Singh Rajput and Shambhavi Upadhyay

Low-Dose CT Image Reconstruction Using Complex Diffusion Regularization 657
Kavkirat Kaur and Shailendra Tiwari

MatConvNet-Based Fast Method for Cervical MR Images Classification 669
Gyanendra K. Verma, Jagdeep S. Lather and Ankur Kaushal

About the Editors

Dr. Nishchal K. Verma (SM'13) is an associate professor at the Department of Electrical Engineering, Indian Institute of Technology Kanpur, India. He obtained his Ph.D. in Electrical Engineering from the Indian Institute of Technology Delhi. He worked as a postdoctoral research fellow at the Center for Integrative and Translational Genomics, University of Tennessee, Memphis, TN, USA, and a postdoctoral research associate at the Department of Computer Science, Louisiana Tech University, Ruston, LA, USA. He was awarded the Devendra Shukla Young Faculty Research Fellowship by the Indian Institute of Technology Kanpur, India (2013–16), and Indo-Canadian SFTIG Award of CAD$6000 for Shastri Faculty Training and Internationalization Program (2017–18). His research interests include cognitive science, big data analysis, bioinformatics, intelligent data mining algorithms and applications, computer vision and image processing, brain–computer–machine interface, machine intelligence, intelligent informatics, smart grids, intelligent fault diagnosis systems, prognosis and health management, soft computing in modelling and control, Internet of things/cyber-physical systems, intelligent agents and their applications, fuzzy logic and controllers, deep learning of neural and fuzzy networks, machine learning algorithms and computational intelligence. He has authored more than 200 research papers and is an IETE fellow. He is currently an editor of the IETE Technical Review Journal, an associate editor of the IEEE Computational Intelligence Magazine and of the Transactions of the Institute of Measurement and Control, UK, and an editorial board member for several journals and conferences.

Dr. A. K. Ghosh is a professor at the Department of Aerospace Engineering, Indian Institute of Technology Kanpur, India. He obtained his Ph.D. in Aerospace from the Indian Institute of Technology Kanpur, India. His areas of expertise are system identification through flight tests using conventional and neural network-based methods, design of aircraft and airborne projectiles, supercavitation and unmanned aerial systems. He has authored more than 100 research papers.

Part I
Intelligent Informatics

Fog Computing-Based Autonomic Security Approach to Internet of Things Applications

S. Prabavathy, K. Sundarakantham, S. Mercy Shalinie and K. Narasimha Mallikarjunan

Abstract Internet of things (IoT) security continues to be a serious problem as the number of connected devices massively increasing day by day. Autonomic security approaches are required for IoT applications to handle the exponentially growing attacks generated from the increasing number of connected devices. With the limited computational and storage capabilities of IoT devices, it is difficult to implement autonomic capability at each device. Hence, fog computing can be used to provide autonomic security characteristics to IoT applications. The proposed fog computing-based security approach has (1) attack forecasting module based on Gaussian process regression model, (2) attack detection module using cognitive approach and correlation analysis, (3) fuzzy-based incident response module to handle the attacks. The experimental results prove the effectiveness of the proposed approach in terms of detection accuracy and detection time.

Keywords Autonomic system · Fog computing · Gaussian regression Cognitive analytics · Fuzzy logic

1 Introduction

IoT has changed the phase of our living style by providing ubiquitous and pervasive environment using network of physical devices embedded with sensors and software [1]. The involvement of connected devices in the intimate spaces generates immense

S. Prabavathy (✉) · K. Sundarakantham · S. M. Shalinie · K. N. Mallikarjunan
Thiagarajar College of Engineering, Madurai, Tamil Nadu, India
e-mail: s.praba.pranauv@gmail.com

K. Sundarakantham
e-mail: kskcse@tce.edu

S. M. Shalinie
e-mail: shalinie@tce.edu

K. N. Mallikarjunan
e-mail: arunkambaraj@tce.edu

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_1

of privacy and security threats. The wireless communication between the connected devices in IoT should meet the security requirement of Internet and sensor network along with the security requirement of IoT to ensure safe and reliable operations [2]. The involvement of evolving revolutionary technologies such as cloud computing, software-defined network, big data analytics in IoT application has increased the vulnerabilities because most of these technologies are at developing stage with non-standard architecture and protocols.

IoT applications need self-protect capability because most of the connected devices are not installed in managed environment [3]. Some IoT applications need the devices to be installed on unmanaged environment such as climatic forecasting, environmental monitoring applications. Another major factor that influences the need for self-protect characteristics is increasing number of connected devices [4]. As the number of connected devices increases in IoT application, the attack surface exponentially increases and it is difficult to handle such massive attack. To protect the IoT application, each connected device has to protect itself from cyberattack by using its own capability. Majority of the IoT devices are simple sensors with limited computational power and memory, where it is not feasible to implement self-protect mechanism. IoT application is made of devices from simple sensors to high-end system; hence, a common security mechanism cannot be applied to all devices. Most of the IoT applications process the data from the connected devices at the cloud. At the same time, implementing self-protect mechanism at cloud side of the IoT application leads to centralized security approach which is subjected to single-point failure and latency in detection. Therefore, the self-protect mechanism of IoT applications are implemented using fog computing (edge computing) [5].

Fog computing provides processing and storage capabilities at the edge with lower latency [6]. Fog nodes connect the end devices of IoT application to cloud computing resources to provide fast, actionable decisions to be made based on vast amount of data generated from the IoT end devices. The self-protect mechanism of IoT application can be implemented at the distributed fog nodes to forecast, detect, and respond to attack at faster rate. In this paper, autonomic approach is used to protect the IoT applications using fog computing. The proposed system is suitable for IoT applications which connects and process the data from end devices at the cloud such as environmental monitoring, weather forecasting, smart home. The major contribution of the paper is

1. Fog computing-based self-protect mechanism for IoT applications is proposed
2. The forecasting module uses Gaussian process regression to predict the attack
3. The detection module uses two kinds of techniques based on the requirement of IoT application
   - Cognition-based approach
   - Correlation-based approach
4. The response module uses fuzzy logic to generate response which can handle the attack efficiently.

The rest of the paper organized as follows. Section 2 provides existing research performed so far in this domain. In Sect. 3, the detailed architecture and functionalities of proposed system is presented. Section 4 presents the experimental setup to evaluate the proposed work. Finally, we concluded the paper by providing possible extensions in proposed autonomic approach to protect IoT applications in Sect. 5.

2 Related Work

IoT applications are made up of smart devices which can sense their surroundings and act accordingly [7]. Hence, self-protection capability can be imparted into these devices to proactively protect themselves without human intervention. Many self-protection systems have been proposed [8], but these are not specific to IoT applications which contain heterogeneous devices from simple sensors to high-end servers. A framework for self-protect mechanism was designed for IoT ecosystem [9]. It is based on centralized control mechanism which is not suitable for large-scale IoT applications that needs a distributed approach with lightweight implementation.

The proposed approach uses Gaussian process regression for forecasting the anomaly. Gaussian process is mainly used for prediction in many applications [10]. In [11], Gaussian process is used to detect the anomaly in smart grid but it does not use fog computing technique with edge intelligence to implement the learning model. Many user behavior analytics-based anomaly detections have been proposed [12]. The proposed system uses a combination of user and behavior analytics to detect anomaly. Similarly, statistical techniques are used for cyberattack detection [13], but do not use the fog computing technology. The fuzzy logic is used in the response module of the proposed system. In existing IoT security, fuzzy logic is mainly used for providing access control [14] to IoT devices and not for detecting or handling the attack.

To the best of our knowledge, this is the first autonomic approach using fog computing to provide self-protect mechanism to IoT application.

3 Fog Computing-Based Self-protect Method

In this section, autonomic approach to protect IoT applications using fog computing is proposed. The IoT application of the proposed method made of end devices, fog nodes, and cloud server. Figure 1 shows the architecture of the proposed approach. In the proposed approach, self-protect method is implemented in the fog node consisting of attack forecasting module, detection module, and response module. The forecasting module uses Gaussian process regression to forecast the cyberattacks in IoT application. The detection module contains two different techniques to detect the attacks (i) cognitive analytics-based detection method and (ii) correlation-based detection method. The detection techniques are chosen based on the application

requirement and capability. Finally, the response module consists of fuzzy logic to identify the most certain response to handle attack scenario. When an attack is detected by detection module or predicted by the forecasting module, alerts are generated and sent to the response module to select correct response to handle the attack.

The self-protect mechanism is implemented at the fog node. The traffic from end devices flows to the cloud server through the fog nodes. Similarly, the data from cloud reaches the end devices through fog node. Since all traffic passes through the fog node, it is efficient to perform the analysis and decision at the distributed fog nodes for large-scale IoT applications. Therefor, self-protection mechanism can also be efficiently performed by the fog nodes.

3.1 Forecasting Module

Forecasting module uses the Gaussian process regression at fog nodes to forecast anomaly in IoT application. The features of data traffic that represents anomaly are considered as random variables to formulate the Gaussian process. Any subset of these random variables can have joint multivariate Gaussian distribution which is

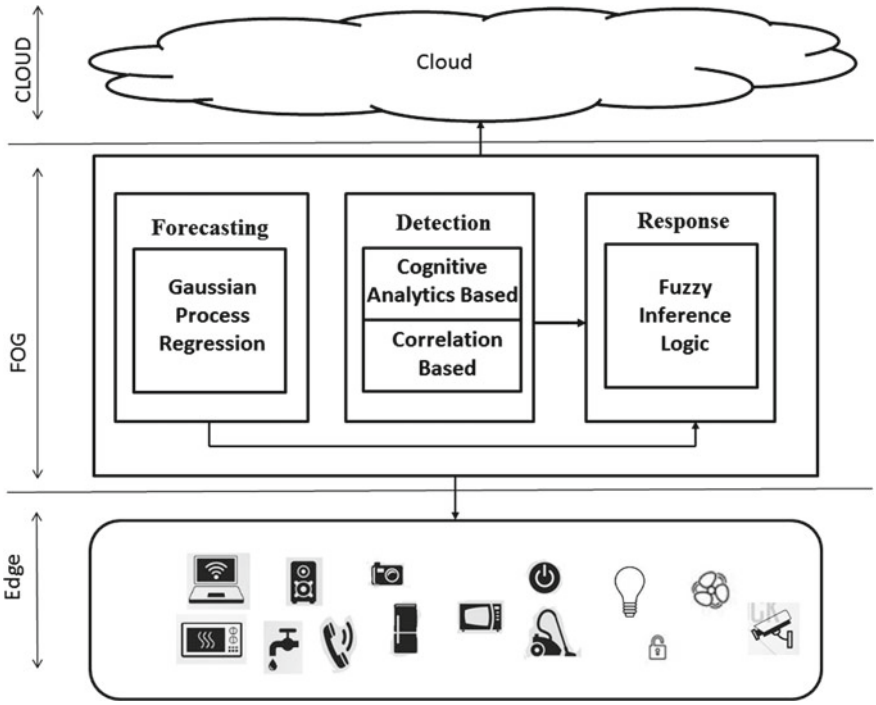


Fig. 1 Architecture of proposed system

represented by mean vector and covariance matrix, whereas Gaussian process is specified by mean function and covariance function. Gaussian process regression is a Bayesian approach which has prior and posterior. Gaussian process is used to define the distributions over the function which can be used as prior belief for Bayesian inference. The prior mean and prior covariance matrix are used for Bayesian inference to compute the posterior distribution which is used to forecast the attacks in the IoT application.

In Gaussian process regression, the output of y of a function f at input x is given as

$$y = f(x) + \varepsilon \tag{1}$$

where ε is the noise represents the randomness in the observation.

$$\varepsilon = N(0, \sigma_{\varepsilon}^{2}) \tag{2}$$

The function $f(x)$ is a Gaussian process distribution

$$f(x) = GP\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \tag{3}$$

The Gaussian distribution *GP* is a distribution over functions defined by mean and covariance function. The mean m(x) is the expected function value at input x

$$m(\mathbf{x}) = E[f(\mathbf{x})] \tag{4}$$

The covariance function $k(\mathbf{x}, \mathbf{x}')$ models the dependence between the function values at different input points $\mathbf{x}$ and $\mathbf{x}'$. The choice of covariance function is very import for optimal prediction. A composite covariance function is used because of the heterogeneous nature of the IoT application. The sum of covariance function considering short-term and long-term trends, the fluctuations in observation length and periodicity. The short-term and long-term trends are represented by two isotropic squared exponential covariance functions. An isotropic quadratic covariance function is used to represent the fluctuations in the observation lengths. The periodicity is represented by smooth periodic covariance function.

3.2 Detection Module

The detection module detects the anomaly using two different techniques (i) cognitive analytics-based detection and (ii) correlation-based detection. The detection technique is based on the IoT application requirement. If the fog nodes of IoT application are resource constrained devices then complex machine learning algorithms cannot be used; therefore, correlation-based techniques are used. If the fog nodes are not resource constrained devices, then cognitive analytics-based detection can be used.

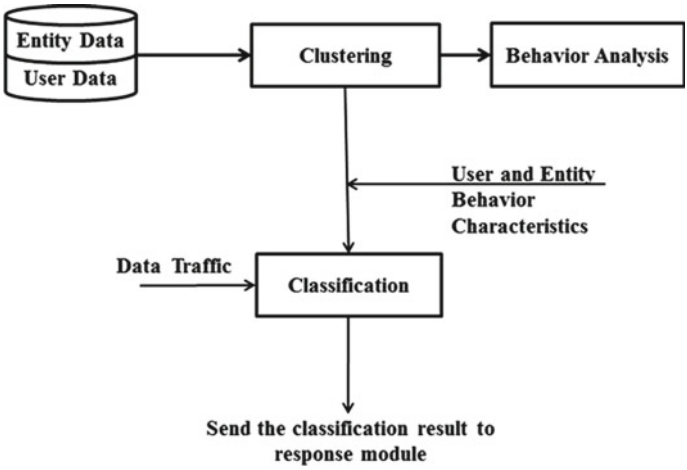


Fig. 2 Cognitive analytics-based detection method

Cognitive Analytics-Based Detection Method

Cognitive analytics provides proactive security system by interpreting data to generate hypothesis based on the knowledge gained from their interactions. The cognitive analytics-based detection mechanism implements a combination of user behavior analytics and entity behavior analytics to proactively identify the anomalies. The user behavior is analyzed based on the user data such as access information, permission, accounts, and resources consumed. The behavior of the entity is analyzed using the data related to network entities. A combination of supervised and unsupervised machine learning algorithms is used to implement the behavior analytics as shown in Fig. 2. Clustering is used to identify important features for behavior analysis and user behavior pattern and entity behavior pattern are inferred from these clusters. The classification algorithm is used to classify the normal and anomaly conditions-based patterns generated by the clustering algorithm.

Cognitive analytics for attack detection involves two steps (i) profiling and (ii) detection. Profiling generates user profiles, entity profile, attack profile, and attacker profile using the behavior analytics. The profile features are used to identify the anomaly by measuring the deviation from the normal profile.

Correlation-Based Detection Method

The abnormality of IoT applications can be detected with simple Pearson correlation analysis [15] at the fog nodes. In some IoT application, highly complex computational techniques cannot be used to detect the anomaly because it was built from resource-constrained devices. Therefore, a simple Pearson correlation is used to detect the abnormalities at fog node. The data traffic between cloud and end devices pass through the fog node which performs correlation analysis with normal stream of packets stored in it. If the resultant correlation coefficient is not within the thresholds, then

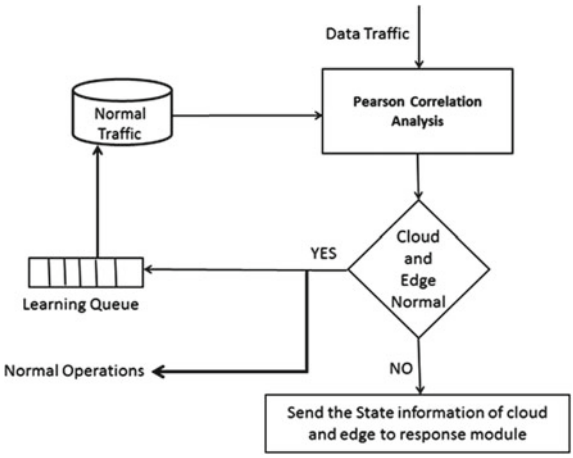


Fig. 3 Correlation-based detection method

it is considered as anomaly. To reduce the false alarm average of three sample's coefficient is used to detect the anomaly. Figure 3 shows the flow of correlation analysis.

3.3 Response Module

Response module generates the highly certain response necessary to handle the attack scenario in the IoT application under two different output from the detection module (i) if cloud or edge devices found abnormal and (ii) if the cloud or edge devices or silent. Based on the result of the detection module, the response is generated using the fuzzy logic controller. When there is uncertainty in the value of input parameters, then fuzzy logic controllers can be used [16]. The result of the detection module gives the state of the cloud and edge devices and it is given as input to the fuzzy inference engine. The output of fuzzy inference system is the incident response for IoT application to handle the attack scenario. Linguistic variables are defined for all input and values are assigned to these linguistic variables. The rule base of fuzzy logic controller stores all the details about input, membership functions, and corresponding output.

The fuzzy inference system in the response module uses Mamdani model consist of two input and one output as shown in Fig. 4. Triangular membership function is used to define the linguistic variables. The two inputs are resulted from the detection module that gives the state of the cloud and edge devices, and the values of two inputs are normal anomaly and silent. The fuzzification process converts the individual input into fuzzy set membership degree to identify highly appropriate response. The rules are written on the rule base based on the result of detection module and the expected response to handle the attack scenario. The centroid method is used for defuzzification to convert the fuzzy output into crisp output (Table 1).

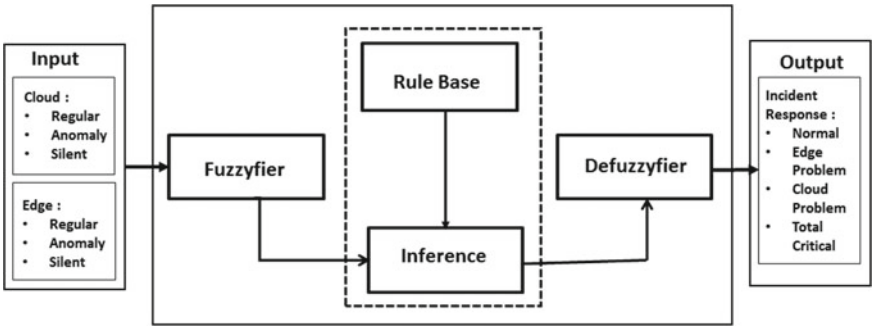


Fig. 4 Fuzzy logic-based incident response

Table 1 Response and actions

Fuzzy response	Action
Normal	Perform normal operations
Cloud problem	Perform basic operations for edge devices without the involvement of cloud Inform the owner and members of the home about the attack through message or mail
Edge problem	Send the attack traffic to the intrusion detection system at the cloud to identify the devices that are being attacked Take actions as per the instructions from cloud
Total critical	Inform the owner and members of the home about the attack through message or mail

4 Experimental Results and Discussion

To demonstrate the proposed system, a simple smart home IoT application is simulated using Cooja [17] simulator as shown in Fig. 5. It uses Contiki [18] operating system on a system with configuration Core i5 3470M processor and 8 GB RAM having GCC 4.9 and Ubuntu 14.04 environment. The simulation uses 6LoWPAN [19] network made of Tmote sky sensor nodes connected to 6BR which acts as fog node. The 6BR is connected to Microsoft Azure cloud service with the computing resource 4x Dual-Core AMD Opteron 2218 @ 2.6 GHz, 8 core, 32 GB RAM, 6 * 146 GB HDD. The 6BR is not a constrained node; therefore, a laptop with DUALCORE processor, 2 GB RAM, 500 GB HDD configuration is used. DDoS attack is generated by generating flooding attack in the 6LoWPAN simulated network to study the performance of the proposed method. As a preliminary work, only correlation-based detection is evaluated in the experiment.

The forecasting module uses Gaussian process regression to predict the attack at faster rate. The prediction time is shown in Fig. 6. The probabilistic prediction is depicted as gray area showing the higher and lower values of energy consumption due to attack. The blue line shows the training part, and the red line shows the testing part

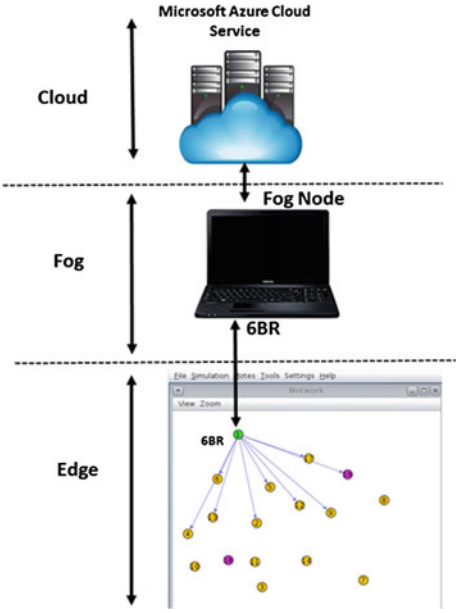
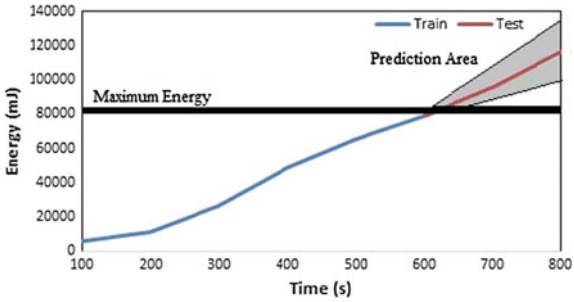


Fig. 5 Experimental setup

Fig. 6 Prediction time of the proposed system



of the forecasting module. The fog node evaluates the prediction time and activates the response module.

The average of Pearson correlation coefficient is computed for the simulated smart home application traffic at the fog node. Fig. 7 shows the lack of correlation between the normal and malicious traffic. The detection accuracy is measured in terms of true positive rate and false positive rate as shown in Fig. 8.

Figure 9 shows the defuzzification of the response module where X-axis and Y-axis denote the cloud and end devices malicious traffic and Z-axis represents response generated based on the rule in the fuzzy inference engine at fog node.

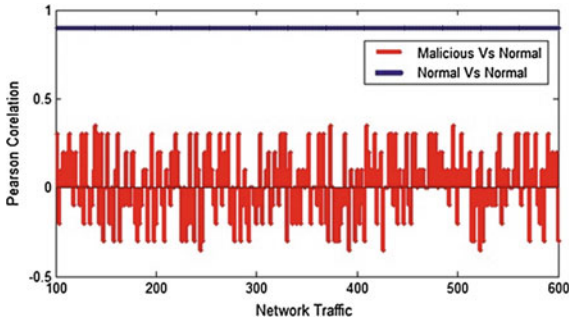


Fig. 7 Correlation analysis for attack detection

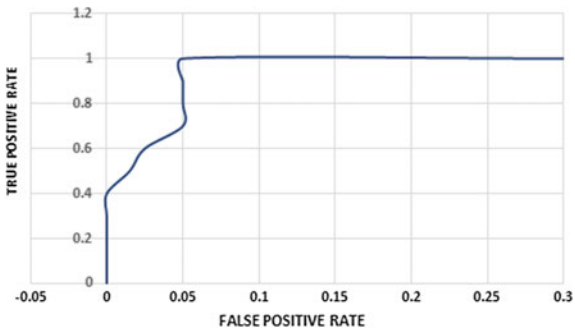


Fig. 8 True positive rate versus false positive rate

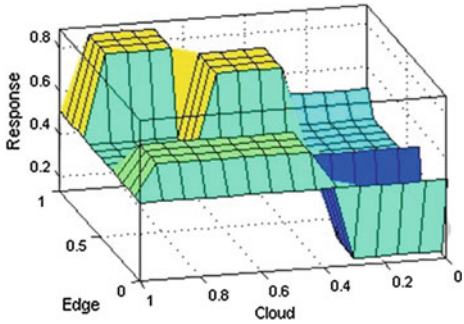


Fig. 9 Defuzzification in response module

To study the efficiency of fog node in terms of latency, the proposed module is implemented in cloud alone without fog nodes. The same experiment is repeated under cloud scenario and delay is measured in terms of average response time. The fog node forecasts, detects, and responds to attacks at faster rate when compared to cloud implementation as shown in Fig. 10.

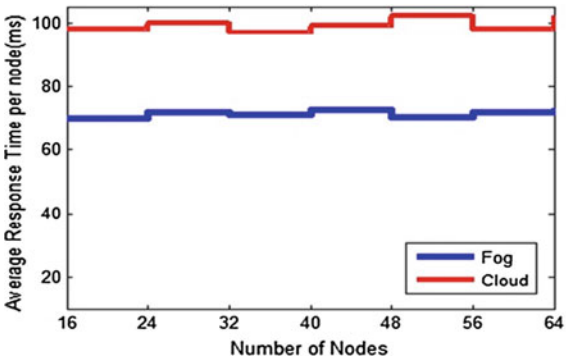


Fig. 10 Average response time of fog and cloud

5 Conclusion

In this paper, an autonomic approach to protect IoT applications using fog computing was proposed. This approach implements the self-protection mechanism at the fog nodes in IoT application to forecast, detect, and respond to the attacks at faster rate. The proposed method uses online linear regression to forecast the attack. Cognitive analytics and correlation analysis are used to detect the attack, and fuzzy logic is used to identify the most appropriate response to handle the attack scenario.

The future work to test the approach in real-time IoT applications is to validate the results and to increase the performance by tuning the proposed approach.

References

1. Wortmann, F., Flüchter, K.: Internet of things. Bus. Inf. Syst. Eng. **57**(3), 221–224 (2015)
2. Sun, X., Wang, C.: The research of security technology in the internet of things. In: Advances in Computer Science, Intelligent System and Environment, pp. 113–119. Springer, Berlin, Heidelberg (2011)
3. Zhao, K., Ge, L.: A survey on the internet of things security. In: 9th International Conference on Computational Intelligence and Security (CIS), pp. 663–667 (2013)
4. Dobson, S., Sterritt, R., Nixon, P., Hinchey, M.: Fulfilling the vision of autonomic computing. Computer **43**(1) (2010)
5. Bonomi, F., Milito, R., Zhu, j., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13–16. ACM (2012)
6. Yi, S., Li, C., Li, Q.: A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data, pp. 37–42. ACM (2015)
7. Xia, F., Yang, L.T., Wang, L., Vinel, A.: Internet of things. Int. J. Commun. Syst. **25**(9), 1101–1109 (2012)
8. Tesauro, G., Chess, D.M., Walsh, W.E., Das, R., Segal, A., Whalley, I., Kephart, J.O., White, S.R.: A multi-agent systems approach to autonomic computing. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 464–471. IEEE Computer Society (2004)

9. Chen, Q., Abdelwahed, S., Erradi, A.: A model-based validated autonomic approach to self-protect computing systems. IEEE Internet Things J. **1**(5), 446–460 (2014)
10. Alecu, T.I., Voloshynovskiy, S., Pun, T.: The Gaussian transform of distributions: definition, computation and application. IEEE Trans. Signal Process. **54**(8), 2976–2985 (2006)
11. Fadlullah, Z.M., Fouda, M.M., Kato, N., Shen, X., Nozaki, Y.: An early warning system against malicious activities for smart grid communications. IEEE Netw. **25**(5) (2011)
12. Stanton, J.M., Stam, K.R., Mastrangelo, P., Jolton, J.: Analysis of end user security behaviors. Comput. Secur. **24**(2), 124–133 (2005)
13. Hurley, N., Cheng, Z., Zhang, M.: Statistical attack detection. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 149–156 (2009)
14. Mahalle, P.N., Thakre, P.A., Prasad, N.R., Prasad, R.: A fuzzy approach to trust based access control in internet of things. In: 3rd International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), pp. 1–5 (2013)
15. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Noise Reduction in Speech Processing, pp. 1–4. Springer Berlin Heidelberg (2009)
16. Harris, J.: An Introduction to Fuzzy Logic Applications, vol. 23. Springer Science & Business Media (2001)
17. Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., Voigt, T.: Cross-level sensor network simulation with cooja. In: 31st IEEE Conference on Local Computer Networks, pp. 641–648 (2006)
18. Dunkels, A., Gronvall, B., Voigt, T.: Contiki-a lightweight and flexible operating system for tiny networked sensors. In: 29th Annual IEEE International Conference on Local Computer Networks, pp. 455–462 (2004)
19. Mulligan, G.: The 6LoWPAN architecture. In: Proceedings of the 4th Workshop on Embedded Networked Sensors, pp. 78–82 (2007)

Self-adaptive Frequent Pattern Growth-Based Dynamic Fuzzy Particle Swarm Optimization for Web Document Clustering

Raja Varma Pamba, Elizabeth Sherly and Kiran Mohan

Abstract Web documents being dynamic and vague, any system to cluster these documents needs to be self-adaptive to these dynamic situations. For this, the system requires the capability of capturing dynamicity. Dynamicity takes into account any updates happening in the search space. If any new potential solution arises, the system needs to identify and reinitialize the particle lists to the newly updated potential solutions. The traditional particle swarm optimization fails in accounts of convergence speed and maintaining diversity. The experimental results for the proposed algorithm show that frequent pattern growth-based dynamic fuzzy particle swarm optimization algorithm performs better than existing conventional approaches in matters of convergence speed and in maintaining diversity.

Keywords Fuzzy · Information retrieval · Particle swarm optimization
Frequent pattern growth

R. V. Pamba (✉)
Department of Computer Science and Engineering, LBS Institute of Technology for Women, Trivandrum, Kerala, India
e-mail: pambaraj@gmail.com

R. V. Pamba
School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kottayam, India

E. Sherly
VRCLC, Indian Institute of Information Technology Management, Kerala, Technopark Campus, Kazhakkoottam, Kerala, India
e-mail: sherly@iiitmk.ac.in

K. Mohan
Payszone LLC LTD, Dubai, UAE
e-mail: me@kiranmohan.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_2

1 Introduction

Recent studies and research in the domain of document clustering are spearheading in the direction of inducing computational intelligence to information retrieval [1–4]. To achieve this goal, researchers are trying to evolve efficient methodologies for imparting dynamic optimization in clustering. The mechanism behind clustering strives to partition set of similar items to one group and dissimilar items to another group, finding its applications in information retrieval, document organization, summarization, and classification [5]. In the domain of web documents clustering especially being unstructured and dynamic in nature, the search space is volatile to any minute fluctuations in the parameters and in the optimal centroids computed. The proposed paper discusses a novel method of dynamic web document clustering that takes into account convergence speed as well as divergence issues which are the common delimiting factors while using particle swarm optimization. The proposed approach not only find optimal cluster centroids using frequent pattern growth (FP Growth) but also help find best optimal solutions as it changes over time [6] using dynamic fuzzy particle swarm optimization. In order to identify optimal solutions which change over time, it is imperative to maintain huge solution diversity using dynamic optimization. Too much diversity would lead to lack in exploitation of best solutions while less diversity would lead to faster convergence. This situation is taken care of by the proposed approach of dynamic frequent pattern growth-based fuzzy particle swarm optimization in providing a balanced search so that the documents can cluster to its relevant entities.

The rest of the paper is structured as follows with Sect. 2 gives an insight on the proposed methodology. While Sect. 3 discusses the results, Sect. 4 deals with the conclusions.

2 Proposed Methodology

The problem definition for dynamic clustering in web document is phrased as below:

$$minimize \quad f(d, \alpha(t)) \tag{1}$$

$$d = (d_1, d_2, \ldots, d_D) \tag{2}$$

Here, we consider $f(d, \alpha(t))$ as the value of fitness function of the particles in the search space solution identified by the frequent pattern growth-based dynamic fuzzy particle swarm optimization. The parameter α at time t captures the change in environment. And the number of dimensions for the search space represented by D. The key objective is to minimize the fitness value of the particle. The fitness function tries to minimize the distances between data objects and its selected cluster centroids, whereas d in Eq. 2 is the particle solutions or vector representation of documents.

Authors in their previous work [7] have developed a reference model that generates particles through frequent pattern growth and tries to cluster those particles to its nearest cluster centroids uisng fuzzy particle swarm optimization. In this paper, emphasis is given on aspects of inculcating a self-adaptive process for the particles to detect any positive changes happenings in their surroundings as discussed in [8] and to adapt to the better position using the proposed algorithm of frequent pattern growth-based dynamic fuzzy particle swarm optimization. Frequent pattern growth algorithm extracts the swarms and particles for fuzzy particle swarm optimization.

2.1 FP Growth

Frequent pattern growth (FP Growth) [9] work in two scans, first scan deals with finding all frequent item sets and second scan creates nodes with updated frequencies for each items being visited in every transaction.

With the frequent patterns received as output from the FP Growth algorithm [10, 11], the proposed approach retrieves all possible combination of terms matching to the user search context effectively. For the selection of frequent patterns, the threshold is set to mean-squared residual error (MSRE) as the criteria. Table 1 gives a reference for the concept translation between the output of FP Growth and the inputs for modified DFPSO as discussed in Sect. 2.6.

2.2 Fuzzy Particle Swarm Clustering

Swarm Intelligence [12–16] is an intelligent paradigm used as an optimization tool. The technique evolved from bio-inspired mechanisms exhibited by fishes, ant, birds, honey bees, fire flies. They exhibit tendencies and urge to follow the behavior of the herd. Fuzzy C-Means (FCM) fails with centroid initialization and local minima hookups. These flaws are resolved with the use of optimization tools like PSO [7, 17–20], ant colony optimization, and genetic algorithms. Among evolutionary algorithms

Table 1 Translating frequent mining concepts to dynamic fuzzy particle swarm clustering

FP Growth concepts	Dynamic fuzzy PSO clustering
Items	Population/Dimension
Item sets	Semantically related terms
Frequent item sets	Swarms for dynamic FPSO
Cluster centroids	Average of each frequent item sets

in practice, the simplest of all in implementation is particle swarm optimization but constrained with issues of convergence speed, dependency on parameters and in maintaining diversity. To cluster web documents, fuzzy particle swarm clustering [21] can be effectively used to capture the vagueness. The matrix, μ_{ij}, is described as follows:

$$\mu_{ij} \in [0, 1] \quad \forall_i = 1, 2, \ldots p \quad \forall_j = 1, 2, \ldots q \tag{3}$$

$$\sum_{j=1}^{q} \mu_{ij} = 1, \quad \forall_i = 1, 2, \ldots p \tag{4}$$

$$J_m = \sum_{j=1}^{q} \sum_{i=1}^{p} \mu_{ij}^m s_{ij} \quad \text{where} \quad s_{ij} = \|o_i - z_j\| \tag{5}$$

$$z_j = \frac{\sum_{i=1}^{p} \mu_{ij}^m o_i}{\sum_{i=1}^{p} \mu_{ij}^m} \tag{6}$$

$$\mu_{ij} = \frac{1}{\sum_{k-1}^{q} \frac{s_{ij}}{s_{ik}}^{\frac{2}{m-1}}} \tag{7}$$

The objects in the search space to be fuzzy clustered are embedded in μ_{ij}. Each matrix has p rows and q columns in which p is the number of data objects and q is the number of clusters. The values at ith row and jth column in the matrix μ_{ij} give the degrees of membership that the particular element has to its respective clusters. In the proposed methodology as shown in Table 1, the parameters for dynamic fuzzy particle swarm optimization are generated by frequent pattern growth algorithm. The frequent item sets generated acts as the swarms or the particles. The cluster centroids are calculated by finding the average of all frequent item sets under every item sets.

The dynamic fuzzy particle swarm optimization (FPSO) begins with parameters generated as in Sect. 2.1. The various parameters derived are, population of particles, initial number of tentative cluster centroids. These cluster cenroid positions indicate the potential solutions for the documents. The position matrix, X, shows the membership degrees between the particles spread across the search space in columns and its related cluster centroids in rows as in Eq. 8. The position matrix is equivalent to the fuzzy matrix. Each column shows how each particle is related to the respective cluster centroids. The position matrix is given below:

$$X = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1c} \\ \vdots & \ddots & \vdots \\ \mu_{n1} & \cdots & \mu_{nc} \end{bmatrix} \tag{8}$$

Further with every iteration to optimize the fitness value, each particle updates its position and velocities until either maximum iteration has reached or until a stage where the position of the particle does not change any longer. The particle fitness value is calculated using a fitness function given as in Eq. 9.

$$F = \frac{C}{J_m}. \tag{9}$$

J_m is computed as in Eq. 5. In every iteration, the documents change its direction and place in the search space by evaluating its own best position i_{best}, neighborhood s_{best} best position and with the fitness value so far obtained.

2.3 Setting Dynamic Environment from Particle Fitness

Major concern is on how best to detect changes and adapt to such changes in the iterations before the optimal solutions are retrieved using fuzzy particle swarm optimization. To achieve this dynamic model, the system should well have all relevant information to guide the search. In dynamic fuzzy particle swarm optimization, the information used to guide the particle to move forward in the search space is through its personal best, i_{best}, and the social best, s_{best}.

The learning probability P_c control factor helps particles to either learn from itself or from others. The value of P_c is set with the particle fitness; in this paper, we have set the fitness minimum the better rather than particle indexes as discussed in [22]. So when the fitness is small, i.e., the particle location is at the best, we set the P_{c_i} of learning other particle i_{best} smaller and set the P_{c_i} from self higher. And in situations when the fitness value evaluated is large, we set the P_{c_i} of learning other particle i_{best} higher and set the P_{c_i} from self-small to account for the dynamic environment.

$$P_{c_i} = \frac{f_i}{S * frequentsets} \tag{10}$$

where f_i is the ith particles fitness value, and frequent sets are the set of population collected from frequent pattern growth approach. S is any constant chosen as 2. So if **i** is large, i.e., fitness is large and hence higher probability for the particle to learn from itself is maintained.

2.4 Maintaining Diversity

Next major concern is for the ways to maintain diversity in this dynamic environment so as to enable the system to identify such changes and self-adapt to any such favorable changes. For this, the proposed model has adopted the mechanism discussed

in [23–26] for addressing the issues of re-initialization of a portion of swarms with dynamically changing inertial weight and changing acceleration coefficients to particles whenever any change is identified. All control parameters used in this study like inertia weight and acceleration coefficients are reset by linearly decreasing over the iterations for optimization. For every predetermined fixed interval changes that are identified, the values are reset to its initial starting values.

As discussed by authors in [27], smaller the inertia weight, ω_i suits best for local search while larger ω_i suits best for global search. Hence, the value of ω_i should decrease during the exploitation process.

$$\omega_i = \omega_{max_i} - (\omega_{max_i} - \omega_{min_i}) * \frac{c_i}{c_{i_{max}}} \tag{11}$$

where $\omega_{max_i}, \omega_{min_i}$ decrease from 0.9 to 0.4, c_i is current iteration, and $c_{i_{max}}$ is the upper limit set of iterations.

2.5 Identifying Change

On every fixed interval of iteration, a particle is randomly chosen through uniform random distribution, $U(0, 1)$. The random particles' fitness value is evaluated. Then, this value is compared to the previous i_{best} and s_{best} so far obtained. If the sample selected has a better fitness value, the position of the particle needs to be modified to the newly chosen random particle as follows:

$$X_i(t) = (1 - \varrho)X_r(t) + (\varrho)X_p(t) \tag{12}$$

where $X_r(t)$ is the random particle position, and $X_p(t)$ is the particle position identified by the frequent pattern growth-based dynamic fuzzy particle swarm optimization. The values of ϱ range from 0 to 1.

2.6 Dynamic Fuzzy Particle Swarm Optimization

Particles are updated and values get reset in a dynamic environment at periodic intervals to find a random position whose fitness values are compared to the fitness value evaluated using frequent pattern growth-based dynamic fuzzy particle swarm optimization. In any of the iteration if it is found that random particle finds a better fitness value than the one controlling the search space then the inertia weight, acceleration coefficient is reset to initial values [13]. The position of the particle is updated with new location changes as follows:

$$X_i(t) = (1 - \varrho)X_r(t) + (\varrho)X_p(t) \tag{13}$$

where $X_r(t)$ is the position for the particle randomly generated from the search space, and $X_p(t)$ is a particle position identified by the frequent pattern growth-based fuzzy particle swarm optimization.The values of ϱ range from 0 to 1.

$$Vel_i(t+1) = w_i \otimes Vel_i(t) \oplus c_1 r_1 \otimes (i_{best}(t) \ominus X_i(t)) \oplus c_2 r_2 (s_{best}(t) \oplus X_p(t))) \tag{14}$$

$$i : 1, 2, \ldots, p \tag{15}$$

$$X_i(t+1) = X_i(t) \oplus Vel_i(t+1) \tag{16}$$

where X_i, Vel_i represent the position and velocity of ith particle. And w_i represent the inertia weight. c_1, c_2 are acceleration coefficients used to influence the particles in their movements in the solution space, p is the number of particles, and r_1 and r_2 are random values in the range [0, 1].

Finally, a stage arises where the position of the particle no more changes with iterations. At this stage, the algorithm converges and retrieves the matrix with values showing the membership of each particle to its fuzzy cluster as shown in Table 3.

3 Experimental Results

Table 2 gives an outline of the data set, 20 Newsgroups which has been chosen for the study. Group 2, 3, 4, 5, 14, 15 represents two specific domains, Computer and Science. A total of 357 messages from two domains is given as input in the form of term-document matrix for clustering. So the documents in the search space need to find their optimal positions closer to their respective optimal cluster centroids identified and captured by the proposed algorithm automatically. Further, the movement of the documents is optimized with dynamic clustering if in cases of any

Table 2 List of items and instances chosen from Newsgroup Data set [28, 29]

Group [29]	Target.Group (#)	Category
2	comp.graphics (54)	comp
3	comp.os.ms-windows.misc (52)	comp
4	comp.sys.ibm.pc.hardware (60)	comp
5	comp.sys.mac.hardware (63)	comp
14	sci.med (63)	sci
15	sci.space (65)	sci

Table 3 Final membership degree values for documents to its clusters

Message ID's	Set1	Set2
1	0.3476109	0.6523891
2	0.2464318	0.7535682
3	0.8447256	0.1552744
4	0.0000000	1.0000000
5	1.000000	0.0000000
6	0.4734385	0.5265615
7	0.0000000	1.0000000
8	0.0000000	1.0000000
9	0.000000	1.0000000
10	0.000000	1.0000000

new updates. Table 3 shows the final membership degrees of given documents to its relevant matching clusters. This clearly depicts and captures the fuzziness and vague nature of web documents showing its presence to multiple clusters.

A small snippet of final fuzzy cluster positions after executing the proposed algorithm on the document sets "Computer" and "Science" is given in Table 3 for the 357 messages chosen from 20 Newsgroup data set. Grouping pattern of documents to its respective clusters and subsequently to their classes is shown in Table 4. Document D_1 in Table 3 shows a fuzzy membership degree value of **0.6523891**. This clearly indicates that the document D_1 belongs to **Set 2**. Set 2 is a subset of class Computer (COMP) as shown in Table 4 with relevant keywords matching to domain of Science(sci) as shown in Table 5.

To check the effectiveness of hybrid algorithm in maintaining diversity and convergence speed, we have chosen K-Means, PSO [30], FCMPSO, and FPFPSO for comparison.

Table 4 Grouping of fuzzy clusters to its relevant classes

Class set	Clusters	Message ID
COMP	C1, C2, C4, C5	{D1, D4}, {D2}, {D4, D7, D10}, {D8, D9}
SCI	C3, C6	{D3}, {D5}

Table 5 FPDFPSO cluster outputs for 20 Newsgroup

Dynamic FPSO clusters	Concepts chain
C3, C6	{sun, file, openwindows, xview, echo, usr, xterm, display, ftp, run}, {motif, file, version, mit, color, font, server, sun, win, screen}, {window, manager, display, event, motif, application, program, widget, win, screen}, {server, lib, xfree, xterm, echo, usr, xterm, display, ftp, run}
C1, C2, C4, C5	{space, nasa, hst, larson, mission, orbit, theory, universe, light, mass}, {nasa, gov, people, energy, apr, earth, ca, jpl, higgins, gary}, {people, pitt, chronic, evidence, body, time, disease, medicine, years, water}, {vitamin, cancer, medical, information, disease, treatment, patients, retinol, good, pms}

Table 6 Comparison of algorithms with proposed FP Growth-based Dynamic FPSO

Algorithm	Avg MSR	No. of frequent patterns	Run time (ms)
K-means	273.45	324	6789
PSO	187.33	523	5462
FCMPSO	156.34	645	5133
FPFPSO	128.21	689	4988
FPDFPSO	121.16	701	4867

4 Conclusion

The effectiveness of the method, frequent pattern growth-based dynamic fuzzy particle swarm optimization (FPDFPSO), was found to be much better in comparison to the traditional approaches of K-means, PSO, FCMPSO, and FPFPSO as seen in Table 6. With minimal convergence speed and minimum mean-squared residual error, the proposed approach retains the maximum of all frequent patterns generated, thereby maintaining diversity. In this paper, we have discussed how the system tries to self-adapt to any changes occurring in the environment by ensembling learning particle to particle fitness rather than just position indexes. So the proposed algorithm resolves the issues of dependency in parameters, local hookups, dynamic environment, managing diversity, and convergence speed seen in conventional particle swarm optimization techniques more efficiently and effectively.

References

1. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recognit. Lett. (Elsevier) **31**(8), 651–666 (2010)
2. Forsati, R., Mahdavi, M., Shamsfard, M., Meybodi, M.R.: Efficient stochastic algorithms for document clustering. Inf. Sci. (Elsevier) **220**, 269–291. https://doi.org/10.1016/j.ins.2012.07.025
3. Manning, C.D., Raghavan, P., Schtze, H.: An Introduction to Information Retrieval. Cambridge University Press (2008)
4. Yates, R.B., Neto, B.R.: Modern Information Retrieval. Addison-Wesley, New York (1999)
5. Aggarwal, C.C., Zhai, C.X.: A survey of text clustering algorithms. Min. Text Data 77–128 (2012)
6. Engelbrecht, A.P.: Computational Intelligence: An Introduction. Wiley (2007)
7. Pamba, R.V., Sherly, E., Mohan, K.: Automated information retrieval model using FP growth based fuzzy particle swarm optimization. Int. J. Comput. Sci. Inf. Technol. **9**(1) (2017)
8. Wang, Z.J., Zhan, Z.H., Zhang, J.: An improved method for comprehensive learning particle swarm optimization. In: IEEE Symposium Series in Computational Intelligence, pp. 218–225 (2015)
9. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min. Knowl. Discov. **8**, 53–87 (2004)
10. Mampaey, M., Tatti, N., Vreeken, J.: Tell me what I need to know: succinctly summarizing data with itemsets. In: 17th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Diego, CA, pp. 573–581 (2011)
11. Yan, X., Cheng, H., Han, J., Xin, D.: Summarizing itemset patterns: a profile-based approach. In: 11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL, pp. 314–323 (2005)
12. Mohammed, A.J., Yusof, Y., Husni, H.: GF-CLUST: a nature-inspired algorithm for automatic text clustering. J. ICT **15**(1), 57–81 (2016)
13. Yang, C., Simon, D.: A new particle swarm optimization technique. In: 18th International Conference on Systems Engineering, ICSEng (2005)
14. Izakian, H., Abraham, A.: Fuzzy C-means and fuzzy swarm for fuzzy clustering problem. Expert Syst. Appl. **38**(3), 1835–1838 (2011)
15. Kennedy, J.F., Eberhart, R.C., Shi, Y., NetLibrary, I.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
16. Yi, W., Yao, M., Jiang, Z.: Fuzzy particle swarm optimization clustering and its application to image clustering. In: Zhuang, Y., Yang, SQ., Rui, Y., He, Q. (eds.) Advances in Multimedia Information Processing—PCM 2006. Lecture Notes in Computer Science, vol. 4261 (2006)
17. Telmo, M., Filho, S., Pimentel, B.A., Souza, R.M.C.R, Oliveira, A.L.I.: Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. Expert Syst. Appl. **42**(17–18), 6315–6328 (2015)
18. Cui, X., Potok, T.E.: Document clustering analysis based on hybrid PSO+Kmeans algorithm. J. Comput. Sci. (Spec. Iss.) 27–33 (2005)
19. Cui, X., Potok, T.E.: A Distributed Flocking Approach for Information Stream Clustering Analysis. Las Vegas, NV, United States (2006)
20. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA (1981)
21. Satapathy, S.C., Patnaik, S.K., Dash, C.D.P., Sahoo, S.: Data clustering using modified fuzzy-PSO (MFPSO). In: Sombattheera, C., Agarwal, A., Udgata, S.K., Lavangnananda, K. (eds.) Multi-disciplinary Trends in Artificial Intelligence, MIWAI 2011. Lecture Notes in Computer Science, vol. 7080 (2011)
22. Mandal, S., Mallik, D., Mandal, D.: Optimal FIR band pass filter design using novel particle swarm optimisation. In: IEEE Symposium on Humanities, Science and Engineering Research, pp. 141–146 (2012)

23. Feng, Y., Teng, G.F., Wang, A.X., Yao, Y.M.: Chaotic inertia weight in particle swarm optimization. In: Second International Conference on Innovative Computing, Information and Control, pp. 475–501. IEEE (2008)
24. Hu, X., Eberhart, R.C.: Adaptive particle swarm optimization: detection and response to dynamic systems. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1666–1670 (2002)
25. Eberhart, R.C., Shi, Y.: Tracking and optimization dynamic systems with particle swarms. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 94–100 (2001)
26. Blackwell, T.M., Bentley, P.J.: Dynamic search with charges swarms. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 19–26 (2002)
27. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: IEEE Proceedings of World Congress in Computational Intelligence, pp. 69–73 (1998)
28. http://qwone.com/~jason/20Newsgroups/
29. http://bayou.cs.ucdavis.edu
30. Lam, Y.-K., Tsang, P.W.M., Leung, C.-S.: PSO-based K-Means clustering with enhanced cluster matching for gene expression data. Neural Comput. Appl. **22**(7–8), 1349–1355 (2013)

A Feature-Reduced Discretized Random Forest Model for Oral Bioavailability Data Classification

Priyanka Shit and Haider Banka

Abstract Oral bioavailability is the measurement of the fraction of admissible drug which reaches the site of action in unchanged form. It is one of the principal pharmacokinetic properties and can be predicted in an early phase of drug discovery and development process. Various computational methods have been used for predicting oral bioavailability of a drug candidate in the literature, which selects some of the compounds from the huge set which are most effective drug candidates and also reduces the cost factor of clinical trials. In this study, we have assigned a class label of all chemical compounds as high (Fractional Absorption F% $\geq$ 50) or low (Fractional Absorption F% $<$ 50) oral bioavailability values. Here, the main aim is to obtain an effective model for classification of oral bioavailability data. In order to achieve this, we have preprocessed oral bioavailable data using Pearson correlation and subset selection as feature reduction methods and data discretization using binning. Discretization is one of the popular data preprocessing technique, which maps continuous data points into discrete values for easy data visualization and improves the performance of classification model. The effectiveness of feature reduction with discretization method for oral bioavailable data has been represented in terms of performance matrices like accuracy percentage, sensitivity, specificity, precision, and negative predictive value. Based on the comparative analysis of the performance of various classification models like artificial neural network (ANN), Bayesian classifier, support vector machine (SVM), K-nearest neighbor with feature-reduced discretized random forest model, we conclude that our proposed model gives better performance over the other compared models.

P. Shit (✉) · H. Banka
Indian Institute of Technology (ISM), Dhanbad, Dhanbad, India
e-mail: priyanka15.ism@gmail.com

H. Banka
e-mail: haider.banka@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_3

Keywords Oral bioavailability · Drug discovery and development · Feature reduction · Discretization

1 Introduction

Drug discovery is the long-term and expensive process, which comprises various important tasks. A large number of ligands are synthesized and tested for selection of effective drug candidate. In this process, drug-likeness is the qualitative notion in which ligands are classified as drug-like or non-drug-like compounds. It can be estimated by different factors and properties in the early phase of drug discovery process using molecular structures. Some features like solubility, logP, potency, physicochemical properties like ligand efficiency, lipophilic efficiency, and some other traditional methods like Lipinski's Rule of Five have been used for predicting drug-likeness of any molecule in the literature. Oral bioavailability is one of the most important factors, which is used to estimate the drug-likeness of any compound [1]. It can be defined as the amount of drug that reaches to the bloodstream in unchanged form to carried out necessary pharmacological and therapeutic effects. Oral bioavailability is related to human intestinal absorption (HIA) kinetics [2, 3], dissolution rate and transportation rate in gastrointestinal fluids across the intestinal membrane which are very important to consider, for calculation of rate and amount of drug absorption in the bloodstream of systematic circulation. Some factors which influence the absorption, permeability of compounds, interactions with transporters, etc. The permeability of any compound decided in terms of the size of the molecule and its capability to make the number of hydrogen bonds. There are some other factors like a number of rotatable bonds, pH dynamics, and mobility which are very important to predict oral bioavailability [4, 5] computationally. Measurement of accurate oral bioavailability is possible by clinical trials but it is very expensive and time-consuming for testing a huge number of compounds. Therefore, to overcome the limitations of high-throughput screening virtual screening is effectively used. In which process, various computational models have been used to perform different tasks.

In recent years, to reduce the time and cost factors various efforts have been made in the area of drug discovery process and consequently, various computational models have been used in the literature like quantitative structure–activity relationship (QSAR), descriptor-based approach. In that case, the predictive efficiency was very low. Due to high false positive rate and small sample size, the performance ability was not productive. However, we have taken relatively large dataset in comparison to previous studies. Some other models, which have been also used in this area, like regression-based models, a model developed by Andrews and coworkers [6], some other classification models were proposed [7], statistical analysis [8–10], QSAR/QSPR models [11, 12], mechanistic models [13, 14], genetic programming [15, 16], machine learning classification [3, 17–19], and other chemoinformatics literatures [20–25].

In this study, the main aim is to propose a model which classify compound as low or high oral bioavailability effectively, for selecting best drug candidates for any target disease. Oral bioavailability is represented in terms of fractional absorption. The range of fractional absorption (F%) value is 0–100, so we have categorized the data into low (fractional absorption value <50) and high (fractional absorption value ≥ 50) oral bioavailable data according to their respective fractional absorption value. If the value of F% is 100% for any drug candidate means the amount of drug intake is goes directly to the bloodstream and completely utilized by the body. We have taken compounds in 2D structures and further represented in terms of features. Features have extracted using different open-source software and tools. To improve the classification accuracy, we have selected an optimal feature set from a huge set using some features selection methods like Pearson correlation and subset selection. For classification, we trained various machine learning techniques like neural networks, Bayes classifier, support vector machine, and k-nearest neighbor. All features have continuous values so we have discretized the features using equal-width binning and we have used random forest model as a classifier in the proposed model. Applying discretization in oral bioavailability dataset, we have got better accuracy with respect to other given models. This paper is organized as follows: Sect. 2 describes the materials and methods used, Sect. 3 describes the feature-reduced discretized random forest model description, and Sect. 4 deals with the results obtained from the simulation study followed by discussions. Finally, Sect. 5 is conclusion.

2 Methods and Materials

In this section, dataset, feature selection techniques, and classification models for the comparative study used in this work have been described.

2.1 Dataset

The dataset collected from ADME/T prediction models and databases [3, 26], which contains 1013 clinically diverse molecules. Each molecule represented in terms of coordinate values with 17 physicochemical properties and some other features have collected from different open-source chemical databases like chemBank, DrugBank, and PaDEL software. Further, the dataset is split into training and test set in which 810 drug candidates have been taken as training set and 203 drug candidates for a test set, respectively.

2.2 Feature Selection

Chemoinformatics deals with a huge number of compounds with different types of chemical properties. Molecular descriptors can be divided into their types based on molecular properties. Some features are similar in nature so to minimize redundancy it is very important to reduce feature space to enhance the prediction performance. Feature reduction [27] provides faster and cost-effective predictive algorithms, and it also helps to understand the underlying process. In order to reduce the no of irrelevant features in this study, we have used Pearson correlation and subset selection method.

Pearson Correlation: Pearson correlation is a basic feature reduction technique which is used to measure the linear dependency or correlation between two features. The outcome of this method lies in the range of -1 to $+1$, both values are inclusive. Where $+1$ is for the positive correlation, -1 is for negative correlation, and 0 is no correlation. The correlation is calculated by the following equation:

$$\gamma = \frac{\sum XY - \frac{(\sum X)*(\sum Y)}{n}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{n}) * (\sum Y^2 - \frac{(\sum Y)^2}{n})}} \tag{1}$$

where X and Y are the features between which correlation is calculated and n is the number of samples.

Subset Selection: The subset selection selects best-performing feature set for any model. It can also divide into wrappers, filters, and embedded methods. The wrapper is more complex as compared to other two. In this method, subset of features is selected by evaluating a model. In the case of filters, feature selection is simpler, it is similar to wrappers in terms of search approach. Embedded methods are specific to any model.

In our work, molecular weight (MW), number of hydrogen bonds count (N_{HBC}), partition coefficient (logP), number of rotatable bonds (N_{RTB}), total polar surface area (TPSA), number of hydrogen bond donor (N_{HBD}), number of hydrogen bond accepter (N_{HBA}), solubility index (logS), shape flexibility index (SFI), and number of rule of five satisfied ($N_{Rule-of-5}$) have been considered as the optimal set of descriptors.

2.3 Classification

There are various computational models have been widely used for classification task in various fields and applications including chemoinformatics, drug discovery, bioinformatics, computational chemistry, structural biology. These methods can handle large dataset and deal with high dimensional data. So these methods have been used for virtual screening of large chemical compounds libraries to classify molecule as active, less active, or inactive, rank according to their activity levels or similarity

matching of molecules using descriptor values of that molecule. In this paper, we have used artificial neural network-based classifiers, support vector machine, probability-based classifiers, lazy learning-based technique for oral bioavailability data classification.

Artificial Neural Network: An artificial neural network is a very common machine learning technique which has a network like structure with some weight. Here, neuron is a basic computation component which associated with activation function. This is supervised learning-based model, and in this process weight is updated in each iteration. Computational complexity depends upon the number of neurons and hidden layers. The number of neurons and hidden layers are varied according to applications.

Naive Bayes Classifier: There are some other approaches for classification methods which are based on probabilistic methods. Naive Bayes is the example of one of this methods. In this method, a prior and posterior probability are calculated for each training data for all features and in the test phase, the new data class value is calculated according to the prior and posterior probability. The data is assigned the class value in which calculated probability is higher. It is very simple technique and used in many applications.

K-Nearest Neighbor Classifier: Computational complexity is basically due to the learning process so to reduce this, many techniques use lazy learning procedure. K-nearest neighbor is an example of a lazy learning process. In this type of learning, all tasks perform in the test phase. It is very simple algorithm but performs very well in various applications.

Support Vector Machine: Support vector machine is a widely used for classification and regression task. It is supervised learning-based model also used with associated learning to analyze data effectively for classification and regression task. The main aim of this algorithm is to increase the separation margin so that data can linearly separable easily.

2.4 Performance Measures

In this study, different machine learning methods have been used for classification. In order to compare all the models, we have used the following performance matrices:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Sensitivity(SN) = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity(SP) = \frac{TN}{TN + FP} \tag{4}$$

$$Precision(PR) = \frac{TP}{TP + FP} \tag{5}$$

$$NegativePredictiveValue(NPV) = \frac{TN}{TN + FN} \tag{6}$$

where, TP = true positives, TN = true negatives, FP = false positives, FN = false negatives.

3 Proposed Model

In this study, our main objective is to select chemical compounds having higher oral bioavailability value. For this, we have learned various machine learning classification models like artificial neural network, naive Bayes, support vector machine, k-nearest neighbor. Before that, the number of features has reduced using two different feature selection techniques. Calculation of Pearson correlation function gives mutual information among features. It is very important to find the correlation among features for decreasing redundancy of features. Further, applying subset selection method, an optimal feature set have selected. Here, we have assigned class labels as low OB or high OB data with respect to fractional absorption value. Data discretization have been commonly used to visualize the data of the given sample effectively. In many applications, it enhances the performance of the classification model. Binning is one of the popular methods of data discretization. Before this, there are two more steps: Molecular descriptor distribution analysis and set cut points and number of bins. All steps have given in Fig. 1 and some of the data distribution analysis have given in Figs. 2 and 3. After binning, we have learned random forest model using training set and validate the model using test set and compared the performance of the proposed model with other classification models using different matrices, which have described in the previous section.

Random forest predictive model uses ensemble learning method which is operated by constructing a multitude of decision trees at training time. Due to ensemble learning, the performance of this model is very high.

4 Result and Discussion

The original dataset, which we have collected, was in SDF form so first all the molecules have been converted and represented in terms of molecular descriptors. An open-source PaDEL software has used for this purpose. A final reduced feature set consists of ten features which have extracted by applying correlation and subset

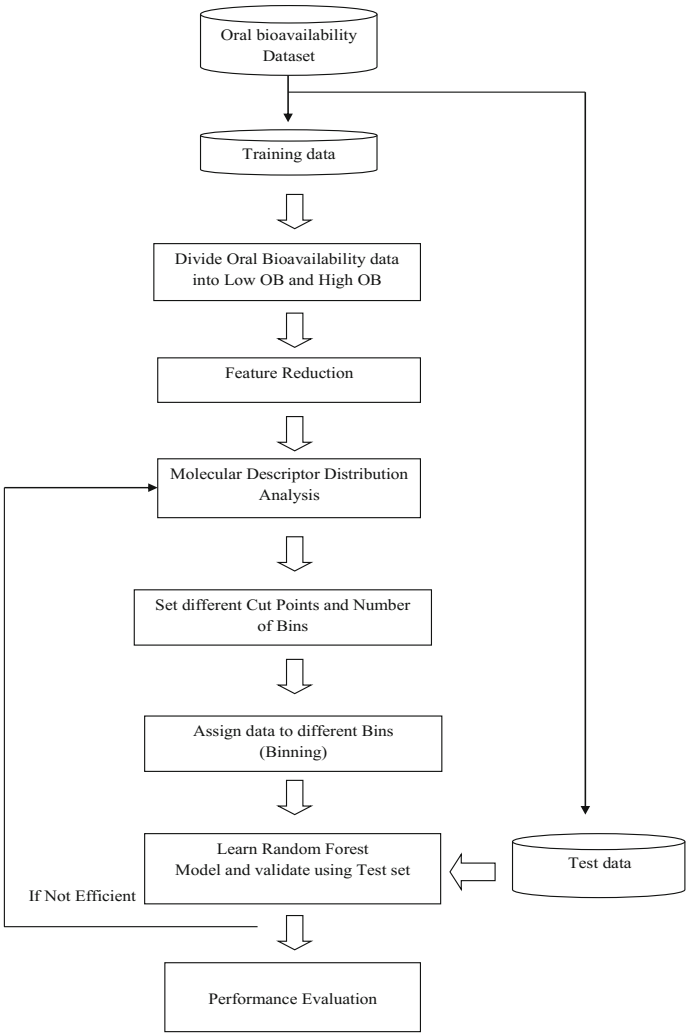


Fig. 1 Feature-reduced discretized random forest classification model

selection method. We have discretized all the features of training data using equal-width binning by setting some cut points for each feature. Then, we have learned the random forest model and validate the result using test set. We have taken 203 molecules for testing, among them 103 molecules belonged to high oral bioavailable data. In this work, we have used WEKA and MATLAB tools for simulation. For artificial neural network, MATLAB software NN Tool has used for implementation. In which, feed-forward back-propagation network and tan-sigmoidal function have selected as network and activation function respectively. Using classification learner, we have trained support vector machine and k-nearest neighbor models. For Bayes

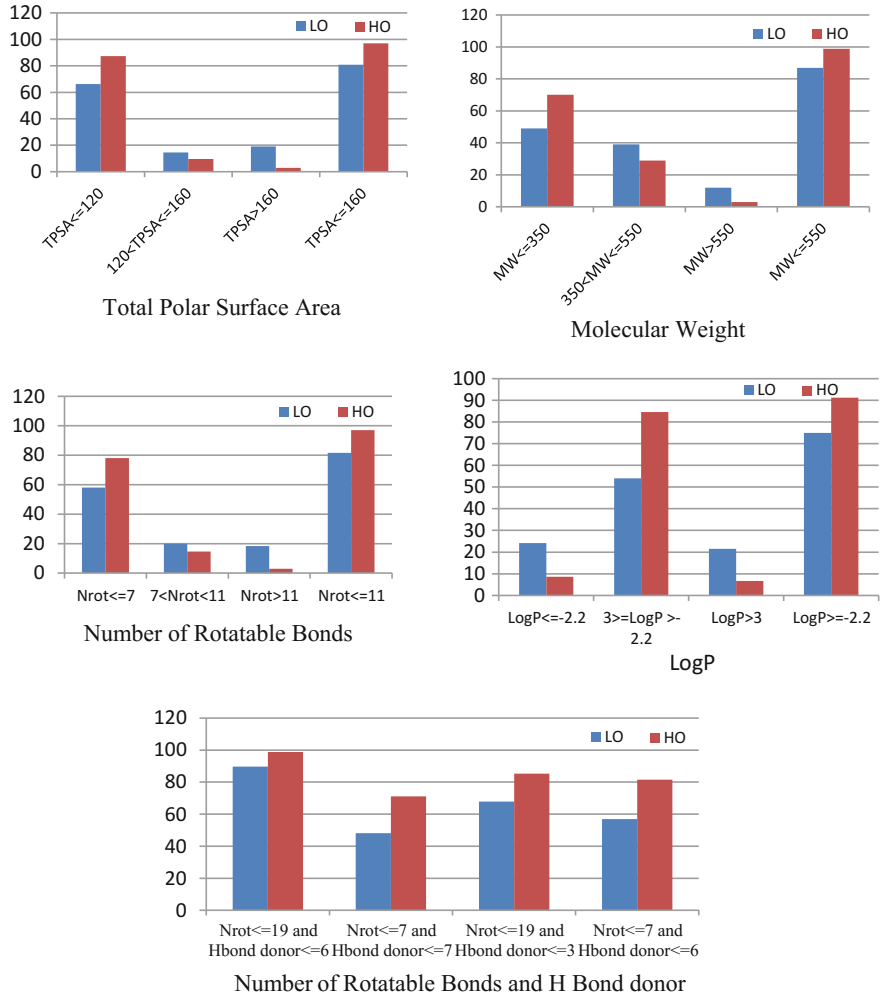
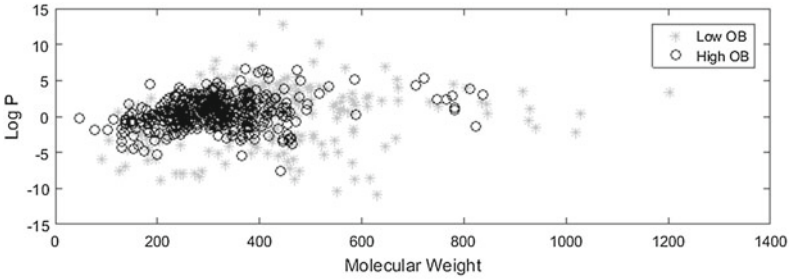
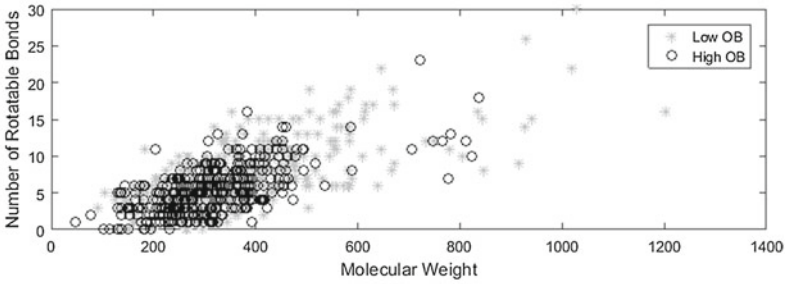


Fig. 2 Data distribution in different cut points of molecular descriptors

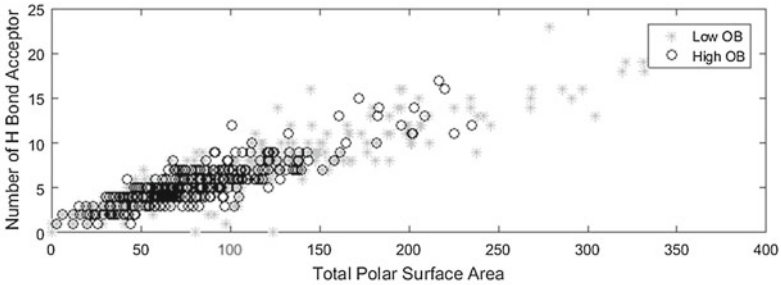
classifier and random forest model, we have used WEKA. Feature reduction and discretization have implemented using MATLAB. After training and testing, we have estimated some performance matrices like accuracy, precision, sensitivity, specificity, and negative predictive value which we have described in earlier section. In this work, for random forest model, we have used bin size 3–9 according to the type of molecular descriptor and data distribution of different cut points. We have got accuracy 80% with discretized random forest model and 74% with support vector machine. Performance of Bayes and neural network model are relatively low, and k-nearest neighbor model has also performed better with accuracy 72%. We have compared all model in terms of different performance matrices as described in Table 1 and from the simulation we



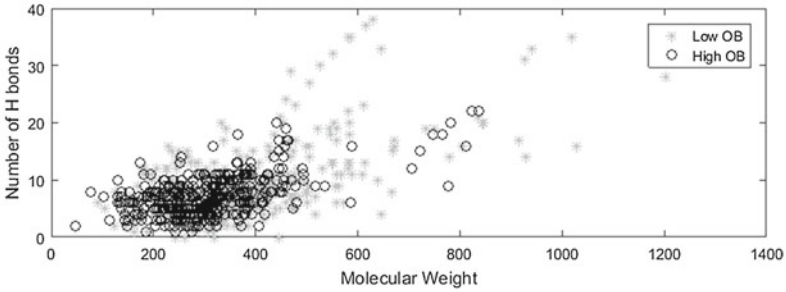
(i) Molecular Weight vs LogP



(ii) Molecular Weight vs Number of Rotatable Bonds



(iii) Total Polar Surface Area vs Number of H bond Accepter



(iv) Molecular Weight vs Number of H Bond

Fig. 3 Distribution of low OB and high OB data in terms of molecular descriptors

Table 1 Performance comparison on terms of accuracy (ACC), sensitivity (SN), specificity (SP), precision (PR), and negative predictive value (NPV)

Classification model	TP	FP	FN	TN	ACC	SN	SP	PR	NPV
Artificial neural network	66	34	30	73	0.68	0.69	0.68	0.66	0.71
Naive Bayes classifier	62	38	32	71	0.66	0.66	0.65	0.62	0.68
K-nearest neighbor	71	29	27	76	0.72	0.72	0.72	0.71	0.74
Support vector machine	71	29	24	79	0.74	0.75	0.73	0.71	0.77
Discretized random forest	76	24	17	86	0.80	0.82	0.78	0.76	0.83

have got Area under curve (AUC) values 0.67, 0.69, 0.72, 0.71, and 0.78 for ANN, Bayes classifier, KNN, SVM, and random forest, respectively, and got maximum accuracy 80% using discretized random forest model.

5 Conclusion

Oral bioavailability prediction is one of the important tasks for selecting suitable drug candidate, which can be done in early phase by virtual screening. This work suggests that machine learning approaches can be effectively used in the early phase of drug discovery and development process. We have classified compounds into low and high oral bioavailability value using various existing classifiers like artificial neural network, naive Bayes, k-nearest neighbor, support vector machine and proposed feature-reduced discretized random forest model. Random forest classifier is a very efficient due to its ensemble learning approach and by using feature reduction and discretization the classification accuracy has improved for this model. Among all the models, the performance of feature-reduced discretized random forest model is significantly better over other compared models.

References

1. Korkmaz, S., Zararsiz, G., Goksuluk, D.: MLViS: a web tool for machine learning-based virtual screening in early-phase of drug discovery and development. PloS One **10**(4), e0124600 (2015)
2. Moda, T.L., Montanari, C.A., Andricopulo, A.D.: Hologram QSAR model for the prediction of human oral bioavailability. Bioorgan. Med. Chem. **15**(24), 7738–7745 (2007)
3. Kumar, R., Sharma, A., Varadwaj, P.K.: A prediction model for oral bioavailability of drugs using physicochemical properties by support vector machine. J. Nat. Sci. Biol. Med. **2**(2), 168 (2011)
4. Veber, D.F., Johnson, S.R., Cheng, H.Y., Smith, B.R., Ward, K.W., Kopple, K.D.: Molecular properties that influence the oral bioavailability of drug candidates. J. Med. Chem. **45**, 261523 (2002)
5. Waterbeemd, H., Gifford, E.: ADMET in silico modeling: towards prediction paradise? Nat. Rev. Drug Discov. **2**, 192204 (2003)

6. Eisenberg, D.M., Davis, R.B., Ettner, S.L., Appel, S., Wilkey, S., et al.: Trends in alternative medicine use in the United States, 1990–1997: results of a follow-up national survey. JAMA **280**, 1569–1575 (1998)
7. Sparreboom, A., Cox, M.C., Acharya, M.R., Figg, W.D.: Herbal remedies in the United States: potential adverse interactions with anticancer agents. J. Clin. Oncol. **22**, 2489–2503 (2004)
8. Lipinski, C.A., et al.: Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Adv. Drug Deliv. Rev. 2**3**(1–3), 3–25 (1997)
9. Lu, J.J., Crimin, K., Goodwin, J.T., Crivori, P., Orrenius, C., Xing, L., et al.: Influence of molecular flexibility and polar surface area metrics on oral bioavailability in the rat. J. Med. Chem. **47**, 61047 (2004)
10. Somogyi, A., Eichelbaum, M., Gugler, R.: Prediction of bioavailability for drugs with a high first-pass effect using oral clearance data. Eur. J. Clin. Pharmacol. **22**(1), 85–90 (1982)
11. Klopman, G., Stefan, L.R., Saiakhov, R.D.: ADME evaluation: 2. A computer model for the prediction of intestinal absorption in humans. Eur. J. Pharm. Sci. **17**(4), 253–263 (2002)
12. Podlogar, B.L., Muegge, I., Brice, L.J.: Computational methods to estimate drug development parameters. Curr. Opin. Drug Discov. Dev. **4**(1), 102–109 (2001)
13. Usansky, H.H., Sinko, P.J.: Estimating human drug oral absorption kinetics from Caco-2 permeability using an absorption-disposition model: model development and evaluation and derivation of analytical solutions for ka and Fa. J. Pharmacol. Exp. Ther. **314**(1), 391–399 (2005)
14. Yoshida, F., Topliss, J.G.: QSAR model for drug human oral bioavailability 1. J. Med. Chem. **43**(13), 2575–2585 (2000)
15. Archetti, F., et al.: Genetic programming for computational pharmacokinetics in drug discovery and development. Genet. Program. Evolvable Mach. **8**(4), 413–432 (2007)
16. Wang, J., et al.: Genetic algorithm-optimized QSPR models for bioavailability, protein binding, and urinary excretion. J. Chem. Inf. Model. **46**(6), 2674–2683 (2006)
17. Turner, J.V., Maddalena, D.J., Agatonovic-Kustrin, S.: Bioavailability prediction based on molecular structure for a diverse series of drugs. Pharm. Res. **21**, 6882 (2004)
18. Frhlich, H., Sieker, F., Wegner, K., Zell, A.: Kernel functions for attributed molecular graphs—a new similarity based approach to ADME prediction in classification and regression. QSAR Comb. Sci. **25**, 31726 (2005)
19. Liu, H.X., Hu, R.J., Zhang, R.S., Yao, X.J., Liu, M.C., Hu, Z.D., et al.: The prediction of human oral absorption for diffusion rate-limited drugs based on heuristic method and support vector machine. J. Comput. Aided Mol. Des. **19**, 3346 (2005)
20. Chen, B., Harrison, R.F., Papadatos, G., Willett, P., Wood, D.J., Lewell, X.Q., et al.: Evaluation of machine-learning methods for ligand-based virtual screening. J. Comp. Aid Mol. Des. **21**, 5362 (2007)
21. Keiser, J., Manneck, T., Vargas, M.: Interactions of mefloquine with praziquantel in the Schistosoma mansoni mouse model and in vitro. J. Antimicrob. Chemother. **66**, 17911797 (2011). https://doi.org/10.1093/jac/dkr178. PMID: 21602552
22. Bielska, E., et al.: Virtual screening strategies in drug design methods and applications. BioTechnol. J. Biotechnol. Comput. Biol. Bionanotechnol. **92**(3) (2011)
23. Korkmaz, S., Zararsiz, G., Goksuluk, D.: Drug/nondrug classification using support vector machines with various feature selection strategies. Comput. Methods Programs Biomed. **117**(2), 51–60 (2014)
24. Arulmozhi, V., Rajesh, R.: Chemoinformatics—a quick review. In: 2011 3rd International Conference on Electronics Computer Technology (ICECT), vol. 6. IEEE (2011)
25. Chonde, S., Kumara, S.: Cheminformatics: an introductory review. In: IIE Annual Conference, Proceedings. Institute of Industrial Engineers-Publisher (2014)
26. Hou, T., Wang, J., Zhang, W., Xu, X.: ADME evaluation in drug discovery. 6. If the oral bioavailability in human can be effectively predicted by simple molecular properties-based rules? J. Chem. Inf. Model. **47**, 460–463 (2007)
27. Mitchell, J.B.O.: Machine learning methods in chemoinformatics. Wiley Interdiscip. Rev. Comput. Mol. Sci. **4**(5), 468–481 (2014)

TelNEClus: Telugu Named Entity Clustering Using Semantic Similarity

SaiKiranmai Gorla, Aditya Chandrashekhar, N. L. Bhanu Murthy and Aruna Malapati

Abstract Semantic similarity plays a significant role in many of natural language processing (NLP) and information retrieval (IR) applications. Most IR methodologies represent the documents using the vector space model (VSM) traditionally known as bag-of-words (BoW) hypothesis. The main disadvantage of BoW is that the grammatical and the structural information of words is not captured. In this paper, we have attempted to cluster named entities (NEs) extracted from Telugu corpus based on semantic similarity. We contend that for this sort of work, more suited VSM is distributional hypothesis which is usually applied for measuring word similarity using word-context matrix. In the word-context matrix, the row vector is words given in the corpus; here, it is a proper noun as most of NEs are proper noun, and column vector is context such as windows of words, grammatical information. The row vector in word-context matrix is constructed in two ways with two different feature sets: The first way is to represent each NE with unique row vector (Row Vector1) without considering different occurrences in a corpus, and the second way is to represent each NE with a set of row vectors (Row Vector2) considering different occurrences in a corpus. For Row Vector1 representation, classical similarity functions like cosine, scalar product, Jaccard can be utilized, but for Row Vector2 representation, we have generalized similarity functions to Sum-of-Sum and Sum-of-Max. Experimentally, Row Vector2 representation enhances the clustering results.

S. Gorla (✉) · A. Chandrashekhar · N. L. Bhanu Murthy · A. Malapati
Birla Institute of Technology and Science, Pilani, Hyderabad Campus, Hyderabad 500078, India
e-mail: p2013531@hyderabad.bits-pilani.ac.in

A. Chandrashekhar
e-mail: snadityac@gmail.com

N. L. Bhanu Murthy
e-mail: bhanu@hyderabad.bits-pilani.ac.in

A. Malapati
e-mail: arunam@hyderabad.bits-pilani.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_4

Keywords Named entities · Vector space model · Semantic similarity
Bag-of-words hypothesis · Distributional hypothesis

1 Introduction

Named entities (NEs) are information elements in text belonging to predefined classes which include the name of the person, location, organization. Recognition and classification of such NEs is known as named entity recognition (NER). Most NER frameworks depend on supervised learning and are trained on a manually annotated corpus with NE classes. However, the primary objective of our work is cluster NEs based on semantic similarity without any supervision about the classes. Measuring the semantic similarity between words or phrases has considerable applications in natural language processing and information retrieval and these are defined using vector space models (VSMs) [1].

The fundamental idea behind VSMs is that it represents documents as a vector in vector space. Factors that are close together are semantically similar to each other. VSMs have various forms, such as a *bag-of-words (BoW) hypothesis*, the *distributional hypothesis*, the *extended distributional hypothesis*, and the *latent relation hypothesis*. In any IR [2] application, the base is BoW hypothesis. The main disadvantage of BoW is that the grammatical and the structural information of words is not captured. In *distributional hypothesis* [3], the words that occur in similar context will have identical meaning and this is applied for measuring word similarity. A word can be represented as a vector in which they are derived from the various occurrences of words in different contexts, such as windows of words [4], grammatical dependencies [5]. Similar row vectors in the word-context matrix indicate identical word meanings.

In this paper, we are keen on clustering named entities namely proper nouns (NNP) based on semantic similarity extracted from Telugu newspaper corpus using word-context matrix. In the word-context matrix, the row vector is words given in the corpus; here, it is NNP, and column vector is context, such as windows of words, grammatical information. The row vector in word-context matrix is constructed in two ways with two different feature sets:

1. Construction of word-context matrix
   - without considering different occurrences of an NE in the corpus (Row Vector1):
     Each NE is represented with a unique vector.
   - considering different occurrences of an NE in the corpus (Row Vector2):
     Each NE is represented as a set of vectors.
2. Feature set
   - windows of words
   - windows of words and grammatical information (part of speech)

The classical similarity functions like scalar product, cosine, Jaccard coefficient can be effectively expanded to deal with this construction. In this paper, these various vector representations and similarity functions are evaluated.

The remainder of this paper is structured as follows: In the next section, we review previous approaches for identifying named entities. Representation of NEs uses word-context matrix is explained in Sect. 3, and experimental results are presented in Sect. 4. Finally, we conclude with a discussion and future work.

2 Background of NER

The named entity recognition task was born in Message Understanding Conference (MUC) (more precisely, it was introduced for MUC-6 in 1995) [6] for identifying and classifying the names of person, location, organizations in a text, as well as numeric expressions including time, date, money, and percent expressions. NER is commonly seen as classification problem which classifies proper noun into broad subgroups [7] or more recently into fine grain groups [8].

Methods for NER are most regularly divided into two streams: **handcrafted rules** and **learning-based approaches** [9]. The earlier NER framework is based on handcrafted rules. It involves the extraction of patterns and using dictionaries which contain lists of NEs. Some of the rule-based systems are FASTUS [10] which uses regular expressions for extracting NEs; LaSIE [11] and LaSIE II [12], which made use of dictionaries of NEs and grammar rules. The major restriction of this framework is that they require human expert of the language and also should have knowledge about the domain. The knowledge and dictionaries are costly to construct and cannot be transferable for different domains. As a result, the focus has moved toward machine learning approaches.

Machine learning is a way to consequently learn to perceive complex patterns and make intelligent choices based on information. Generally, they are classified into **supervised**, **semi-supervised**, and **unsupervised** learning. In supervised learning, an annotated training dataset is given as input. A classification algorithm is used to generalize from these inputs and identify sets of rules that can be applied to a new document to identify named entities in it. Several distinct approaches to classification algorithms are developed such as support vector machine [13], conditional random field [14], maximum entropy model [15], hidden Markov model [16], Naïve Bayes [17], and decision trees [18]. The main drawback of this approach is their crucial dependence on the availability of large, representative, and high-quality annotated training data. Typically, such data are created manually by experts. This makes the creation of data an expensive, time-consuming, and error-prone task. The major difference between supervised and semi-supervised learning is that semi-supervised learning uses both labeled and unlabeled data. Different methods in semi-supervised learning are bootstrapping or self-training [19], co-training [20].

Unsupervised learning approach learns from unannotated data and automatically infers occurrences of NE. The unsupervised learning approaches are clustering, using lexical resources [21] (e.g., WordNet), on lexical patterns [22] and distribution statistics on the unannotated data. The work on unsupervised learning is clustering named entities using generative model [23] which identify named entity internal structure and cluster-related words based on its role. The model uses features from the named entity itself and its syntactic context, and co-reference information. The other work is based on word embedding [24] as neural network-based language models generate highly informative vector representations. The other unsupervised work on clustering is named entities in relation identification [25].

The earlier work on named entity recognition for Telugu language is based on supervised learning. In this paper [26], CRF-based noun tagger was built and then they developed rule-based NER system for Telugu which mainly focuses on identifying name of person, location, and organization. They also developed CRF-based NER system for Telugu. In this paper [27], they developed CRF-based NER system. In this paper [28], hybrid approach, i.e., combination of rule-based and CRF model, was developed.

Telugu, like other Indian languages, annotated corpus, dictionaries are not available. So, we have explored clustering method for Telugu named entities based on semantic similarity. The named entity clusters can be used as a dictionary or gazetteers of names, which help in improving the performance of supervised learning approaches.

3 Representation of NEs Using Word-Context Matrix

In our task, we concentrate only on proper nouns (NNP) from Telugu newspaper corpus. The corpus is part of speech (PoS) tagged using TnT tagger [29], and then NNPs are extracted based on the tag. In order to cluster similar NEs together, the NNPs are represented using word-context matrix. In the remaining section, we explain features used for word-matrix construction.

3.1 Features

We have considered both windows of words and PoS features. These are based on uni-gram, bi-gram, and trigram collected in a window (W) of 5, i.e., 4 (N) tokens before and after each NNP occurrence in the sentence. An example of collected n-grams is given in Fig. 1.

Sentence	
పీసీసీ అధ్యక్షుడిగా రాష్ట్ర మంత్రిగా పలుసార్లు పనిచేసి అనుభవమున్న డీఎస్‌కు టీఆర్ఎస్ సర్కారు ప్రతిష్ఠాత్మకమైన సలహాదారు పదవినిచ్చింది . pIsIsI adhyaxuDigA rAshTa maMtrigA pallusArlu panicEsi anubhavamunna DIesku TIAres sarkAru pratishThAtmakamaina salahAdAru padavinicciMdi.	
NNP	**n-gram feature**
పీసీసీ (pIsIsI)	అధ్యక్షుడిగా రాష్ట్ర మంత్రిగా | రాష్ట్ర మంత్రిగా పలుసార్లు adhyaxuDigA rAshTa maMtrigA | rAshTa maMtrigA pallusArlu
డీఎస్‌కు (DIesku)	మంత్రిగా పలుసార్లు పనిచేసి | పలుసార్లు పనిచేసి అనుభవమున్న maMtrigA pallusArlu panicEsi | pallusArlu panicEsi anubhavamunna
టీఆర్ఎస్ (TIAres)	పలుసార్లు పనిచేసి అనుభవమున్న | సర్కారు ప్రతిష్ఠాత్మకమైన సలహాదారు | ప్రతిష్ఠాత్మకమైన సలహాదారు పదవినిచ్చింది pallusArlu panicEsi anubhavamunna | sarkAru pratishThAtmakamaina salahAdAru | pratishThAtmakamaina salahAdAru padavinicciMdi

Fig. 1 N-gram features for NNP N = 4, W = 5

3.2 Weighting the Elements

The idea of weighting is to provide greater weight to sudden elements and much less weight expected element. In information theory, a sudden element has a higher information than a less weighted element. We have used *tf-idf* (term frequency $\times$ inverse document frequency) weighting scheme. The term frequency of an element is the number of times the element appears in the given document. The tf is usually calculated using the following formula:

$$tf = 1 + log(tf) \tag{1}$$

where tf stands for the number of times the element appears in the document. The inverse document frequency gives lesser weight to most frequent words, such as stop words. The idf is calculated using the following formula:

$$idf = log(N/df) \tag{2}$$

where N is the total number of documents and df is the frequency of documents that contain the element. In this task, N is the total number of times NE occurred in the corpus and df is the frequency of NE that is surrounded by the term.

3.3 Formulation of Word-Context Matrix

In general, in word-context matrix, the context is given by words. Similar row vectors in the word-context matrix indicate similar word meanings. In this work, similar row vectors indicate similar NEs. The row vector in word-context matrix is represented in two ways:

- **without considering different occurrences of an NE in the corpus (Row Vector1):**
  Each NE is represented with a unique vector, but the comparison of two NNP cannot be made at the occurrence level. Figure 2 is an example for above representation. NNP_{11}, NNP_{21}, NNP_{31} are different NNPs in a corpus, but NNP_{11} and NNP_{12} are same NNP but at different occurrences.
- **considering different occurrences of an NE in the corpus (Row Vector2):**
  Each NE is represented as set of vectors. Different occurrences of an NNP are added and the comparison of two NNPs at the occurrence level is possible.
  Figure 3 is an example for above representation. NNP_1 and NNP_2 are set vectors. The representation is defined in Eqs. 3 and 4.

$$NE_1 = \{NNP_{11}, NNP_{12}, \ldots, NNP_{1i}, \ldots, NNP_{1n}\} \tag{3}$$

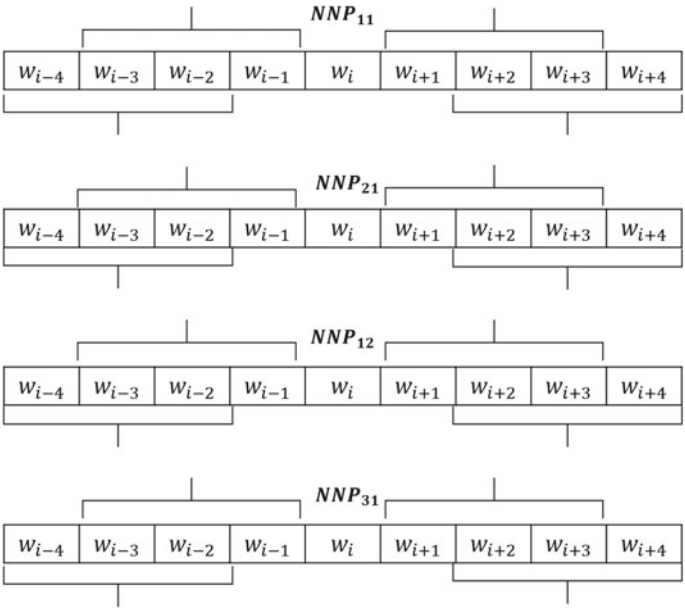


Fig. 2 Row Vector1 without considering different occurrences of an NE

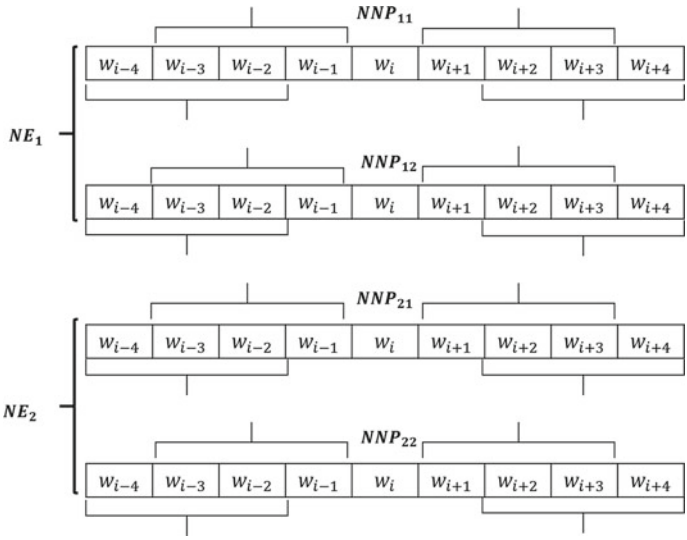


Fig. 3 Row Vector2 considering different occurrences of an NE

$$NE_2 = \{NNP_{21}, NNP_{22}, \ldots, NNP_{2i}, \ldots, NNP_{2n}\} \tag{4}$$

where NE_1 and NE_2 are NNPs and n is the number of occurrences of NE_1 and NE_2 in the corpus. NNP_{1i} and NNP_{2i} are the ith occurrences of NE_1 and NE_2 in the corpus.

4 Similarity Functions

The classical similarity functions like cosine similarity, scalar product, Jaccard Coefficient work well with Row Vector1 representation. In order to use those similarity functions for Row Vector2 representation, we need to generalize them. The simplest way is to sum all the similarities computed from all the possible combinations of vector comparison from the two BoV, for instance, based on the work [30]. There are different ways to generalize the similarity function, such as Sum-of-Sum or Sum-of-Max. The example is shown in Fig. 4 and **W** represents weight. In this paper, we use both Sum-of-Sum and Sum-of-Max definitions which are formulated in Eqs. 5 and 6.

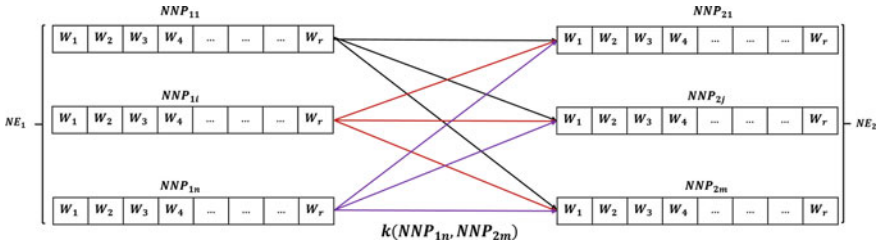


Fig. 4 Calculating similarity

Where $NE_1 = b_{11}, b_{12}, \ldots, b_{1i}, \ldots, b_{1n}$ and $NE_2 = b_{21}, b_{22}, \ldots, b_{2j}, \ldots, b_{2s}$. In Eqs. 5 and 6, k can be any standard similarity function and n, m are the number of different NNPs in NE_1s and NE_2s, respectively.

$$Sim_{SS}(NE_1, NE_2) = \sum_{i=1}^{n} \sum_{j=1}^{s} k(NNP_{1i}, NNP_{2j}) \tag{5}$$

$$Sim_{SM}(NE_1, NE_2) = \sum_{1=1}^{n} \max_{j=1}^{s} k(NNP_{1i}, NNP_{2j}) \tag{6}$$

5 Markov Clustering

In our task, clustering of proper nouns (NNP) can be considered as a graph clustering, in which each node in the graph is considered as NNP and an edge is the relation between two NNPs. We have used Markov clustering algorithm (MCL) as it is a graph clustering algorithm [31]. The main advantage of MCL over other classical clustering algorithms is that it does not require to specify the number of clusters. Characteristic clusters in a graph are characterized by the nearness of numerous edges between the individuals of that cluster, and one anticipates that the number of higher-length (longer) ways between two subjective hubs in the cluster is tall. In specific, this number ought to be high, relative to nodes lying in distinctive common clusters. A distinctive point on this is that arbitrary strolls on the graph will occasionally go from one characteristic cluster to another.

The MCL finds clusters by a mathematical bootstrapping procedure. The technique deterministically computes the probability of random walks, and it uses two operators that transform one set of probability to other. This algorithm simulates random walks within a graph via alternation of two operators known as Expansion and Inflation. The Expansion operator does the matrix squaring, and Inflation takes the Hadamard power of a matrix.

6 Experimental Results

In this section, we initially report evaluation metrics for similarity functions, the data for experiment, and then the outcomes with various similarity functions are presented.

6.1 Evaluation Metrics

Evaluation metrics for clustering such as Rand measure [32], it computes how similar the clusters are the benchmark classifications. However, this metric is exceptionally separating, some of the time being over-optimistic, particularly when the numbers of individuals in each cluster are generally little. To the contrary, Adjusted Random Index (ARI) is an adjusted-for-chance form of Rand measure. It is chosen as the main evaluation metric in this paper.

The ARI computes a similarity measure between two clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusters. The ARI [33] is defined as follows. Given a set of n elements $S = \{O_1, \ldots, O_n\}$ and two partitions of S to compare, $U = \{u_1, \ldots, u_r\}$ and $V = \{v_1, \ldots, v_c\}$, where n_{ij} is the number of common objects between two partitions u_i and v_j.

The Adjusted Random Index is defined in equation

$$ARI = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} \tag{7}$$

where

$$Index = \sum_{ij} \binom{n_{ij}}{2}$$

and

$$ExpectedIndex = \frac{\sum_i \binom{a_{i2}}{2} \sum_j \binom{b_{j2}}{2}}{\binom{n}{2}}$$

and

$$MaxIndex = \frac{1}{2}(\sum_i \binom{a_{i2}}{2} + \sum_j \binom{b_{j2}}{2})$$

Table 1 NE classes in ground truth

Cluster label	N	Of total (%)
Player	893	72
Town	134	11
Team	113	9
Other	30	2
Country	35	2
Stadium	18	1
Sports	12	1

6.2 Data

In this experiment, we use a Telugu sports data extracted from news Web sites. The data consist of 1002 sentences, 13,425 words, and 1235 occurrences of NNPs (325 unique NNP). In order to build a ground truth, we manually cluster the NNPs. It resulted in seven ground clusters, including player name, country, town, team name, sports name, which are listed in Table 1.

6.3 Results

In our experiments, we evaluated two different row vector representations as explained in Sect. 3.3 and Figs. 2 and 3 for NNP clustering task. We have conducted two different experiments.

1. construction of **Row Vector1** and **Row Vector2** using windows of words (Feature Set1) with window of 5.
2. construction of **Row Vector1** and **Row vector2** using windows of words and PoS information of words (Feature Set2) with window of 5.

For these two representations, we use scalar product, cosine, and Jaccard coefficient similarity functions with Markov clustering algorithm. For the three different similarity functions, we report the results for Row Vector1 representation. For Row Vector2 representation, the similarity measure for each vector is combined with Sum-of-Sum (SS) and Sum-of-Max (SM). The results are presented in Tables 2 and 3.

The results show that **Row Vector2** outperforms **Row Vector1** in every case. The **Row Vector2** suits better for this type of tasks. The results are listed in Tables 2 and 3 which shows that SS similarity made better results than SM. Edabat et al. [34] shows that for French corpus, windows of word feature were giving better results, but for Telugu, we explored that window of words and PoS information were giving better results than only window of words feature.

Table 2 Similarity function comparison with Sum-of-Sum and Sum-of-Max, in terms of ARI (%) using Feature Set1

Similarity function	Row Vector1	Row Vector2 (SS)	Row Vector2 (SM)
Cosine	8.26	48.11	47.15
Jaccard	9.43	46.59	45.89
Scalar product	42.04	44.12	43.45

Table 3 Similarity function comparison with Sum-of-Sum and Sum-of-Max, in terms of ARI (%) using Feature Set2

Similarity function	Row Vector1	Row Vector2 (SS)	Row Vector2 (SM)
Cosine	9.56	60.45	53.13
Jaccard	10.23	49.56	48.45
Scalar product	58.14	58.14	55.67

6.4 Error Analysis

The **Row Vector2** representation is a good representation for clustering of named entities. We calculated the errors in final clustering results. In order to find the errors for each class in ground truth, and we also explored what are the NNP that are not clustered.

In order to explore the error, we calculated precision, recall [35] for each NNP on the clusters. For combination cosine similarity combined with Sum-of-Sum, the precision, recall, and F-measure are reported in Table 4. The best F-measure is for the player name, class which is the most important class in the corpus. The most confused class was the country class with player name because most often player names are followed by country class than contextual features will be similar between country class and player name. The second best is town class with a recall of 87.95% which is the highest of all the class which means that all NNPs in this class are more

Table 4 Precision, Recall, and F-measure

Class	Precision	Recall	F-measure
Player	78.84	84.52	81.58
Town	65.12	87.95	74.83
Team	35.12	38.45	36.71
Other	18.12	20.38	19.18
Country	48.75	19.68	28.04
Stadium	10.25	45.23	16.71
Sports	7.12	49.47	12.44

in number in a cluster. The class evaluation for sports is the most difficult class to cluster because the n-grams surrounded by sports are scattered in the corpus.

7 Conclusion and Future work

We have presented named entity recognition problem as a clustering task. In an evaluation of the Telugu corpus, the approach achieves very promising results. We have explored two important aspects related to the NNP clustering task: (1) Window of words and PoS information features play an important role, and (2) difference occurrences of an NNP are also important for clustering. In contrast to other approaches to named entity recognition, our approach offers several advantages: No annotation is required, the distance metrics are easily extended to our representation, and the Markov clustering approach gives an advantage as it does not require to specify the number of clusters.

In our approach, we only used maximum and sum of aggregate as similarity measures, and both these measures can be extended to OR logical operations, fuzzy operations. We would also like to explore word embedding techniques for constructing vectors from a raw corpus and apply different clustering methods for named entity clustering task for Telugu.

References

1. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**, 141–188 (2010). https://doi.org/10.1613/jair.2934
2. Salton, G., Wong, A., Yang, C.-S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
3. Harris, Z.S.: Distributional structure. Word **10**(2–3), 146–162 (1954)
4. Lund, K., Burgess, C.: Producing high-dimensional semantic spaces from lexical co-occurrence. Behav. Res. Methods Instrum. Comput. **28**(2), 203–208 (1996)
5. Lin, D., Pantel, P.: DIRT@ SBT@ discovery of inference rules from text. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2001)
6. Grishman, R., Sundheim, B.: Design of the MUC-6 evaluation. In: Proceedings of a Workshop on Held at Vienna, Virginia, 6–8 May 1996. Association for Computational Linguistics (1996)
7. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics (2003)
8. Fleischman, M., Hovy, E.: Fine grained classification of named entities. In: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics (2002)
9. Sarawagi, S.: Information extraction. Found. Trends Databases **1**(3), 261–377 (2008)
10. Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., Kameyama, M., Martin, D., Tyson, M.: SRI International FASTUS system: MUC-6 test results and analysis. In: Proceedings of the 6th

Conference on Message Understanding, pp. 237–248. Association for Computational Linguistics (1995)
11. Gaizauskas, R., Humphreys, K., Cunningham, H., Wilks, Y.: University of Sheffield: description of the LaSIE system as used for MUC-6. In: Proceedings of the 6th Conference on Message Understanding, pp. 207–220. Association for Computational Linguistics (1995)
12. Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., Wilks, Y.: University of Sheffield: description of the LaSIE-II system as used for MUC-7. In: Proceedings of the Seventh Message Understanding Conferences (MUC-7) (1998)
13. Isozaki, H., Kazawa, H.: Efficient support vector classifiers for named entity recognition. In: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics (2002)
14. Arnold, A., Nallapati, R., Cohen, W.W.: Exploiting Feature Hierarchy for Transfer Learning in Named Entity Recognition. ACL (2008)
15. Chieu, H.L., Ng, H.T.: Named entity recognition: a maximum entropy approach using global information. In: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics (2002)
16. Collier, N., Nobata, C., Tsujii, J.: Extracting the names of genes and gene products with a hidden Markov model. In: Proceedings of the 18th Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics (2000)
17. Mohit, B., Hwa, R.: Syntax-based semi-supervised named entity tagging. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics (2005)
18. Finkel, J.R., Manning, C.D.: Nested named entity recognition. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1. Association for Computational Linguistics (2009)
19. Knopp, J.: Extending a Multilingual Lexical Resource by Bootstrapping Named Entity Classification Using Wikipedia's Category System, pp. 35–43 (2011)
20. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999)
21. Alfonseca, E., Manandhar, S.: An unsupervised method for general named entity recognition and automated concept discovery. In: Proceedings of the International Conference on General WordNet (2002)
22. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. Artif. Intell. **165**, 91–134 (Essex: Elsevier Science Publishers) (2005)
23. Elsner, M., Charniak, E., Johnson, M.: Structured generative models for unsupervised named-entity clustering. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics (2009)
24. Passos, A., Kumar, V., McCallum, A.: Lexicon infused phrase embeddings for named entity resolution. arXiv:1404.5367 (2014)
25. Bollegala, D.T., Matsuo, Y., Ishizuka, M.: Relational duality: unsupervised extraction of semantic relations between entities on the web. In: Proceedings of the 19th International Conference on World Wide Web. ACM (2010)
26. Srikanth, P., Murthy, K.N.: Named Entity Recognition for Telugu. IJCNLP (2008)
27. Shishtla, P. et al.: Experiments in Telugu NER: A Conditional Random Field Approach. IJCNLP (2008)
28. Humera Khanam, M., Khudhus, Md. A., Prasad Babu, M.S.: Named entity recognition using machine learning techniques for Telugu language. In: 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE (2016)
29. Reddy, S., Sharoff, S.: Cross language POS taggers (and other tools) for Indian languages: an experiment with Kannada using Telugu resources. In: Proceedings of the Fifth International Workshop on Cross Lingual Information Access, pp. 11–19 (2011)

30. Gosselin, P.H., Cord, M., Philipp-Foliguet, S.: Kernels on bags of fuzzy regions for fast object retrieval. In: Image Processing, ICIP, vol. 1, pp. 1–177 (2007)
31. Van Dongen, S.: Graph clustering by flow simulation. Ph.D. thesis, University of Utrecht, Utrecht, The Netherlands (2000)
32. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
33. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)
34. Ebadat, A.-R., Claveau, V., Sbillot, P.: Semantic clustering using bag-of-bag-of-features. In: CORIA-COnfrence en Recherche d'Information et Applications (2012)
35. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval, vol. 1, no. 1. Cambridge University Press, Cambridge (2008)
36. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvist. Investig. **30**(1), 3–26 (2007)

Convolutional Neural Network with Stacked Autoencoder for Kernel Initialization

Vikas Singh, Anirudh Swaminathan and Nishchal K. Verma

Abstract In recent years, convolutional neural networks have gained popularity in the area of image processing, machine translation, speech recognition, object detection, and many other tasks. The data generated in all these areas are very large, and there are a large number of samples along with a large number of attributes. Areas like bioinformatics have a large amount of data, but face the problem of a small number of samples with a large number of attributes. In all these applications, initialization plays a better role for better generalization of the network. In this work, we have proposed a novel approach for kernel initialization in which the weights learned by each autoencoder hidden layer acts as the initial kernel (filter) weight of each convolutional neural network layer. The result of the proposed approach is compared with random initialization of the kernel weights. The results show that the proposed method performs comparably to random initialization.

Keywords Machine learning · DL · CNN · AE · SAE

1 Introduction

In the era of big data, data is generated in every domain. Handling such large amount of data using classical machine learning is difficult. Deep learning, which is an advancement in machine learning, is able to handle such large and complex data.

V. Singh (✉) · N. K. Verma
Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: vikkyk@iitk.ac.in

N. K. Verma
e-mail: nishchal@iitk.ac.in; nishchal.iitk@gmail.com

A. Swaminathan
National Institute of Technology Trichy, Trichy, India
e-mail: aniswami97@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_5

The term deep learning was coined in the early 2000s but it was popularized in the year 2006 when Hinton et al. [1–3] proposed a deep neural network (DNN) in which they derived a fast, greedy algorithm that learnt deep networks one layer at a time. Vincent et al. [4] built a DNN by layer-wise stacking of a denoising autoencoder to learn useful representations with local denoising criterion. Baldi [5] had discussed autoencoders which will play a fundamental role for unsupervised feature learning and form a deep architecture. Basically, autoencoders were proposed by Hinton in the 1980s and the PDP group of Rumelhart et al. [6] to address the difficulty of "back propagation". Sevakula and Vikas, et al. [7–10] have also used stacked autoencoder as DNN for feature learning for cancer data. The generalization capability of machine learning model for various case studies are also presented in [13–15].

Inspired by the structure of the visual system, a convolutional neural network (CNN), which is a type of DNN, was theorized. The work of Hubel and Wiesel [16] on cats primary visual cortex in 1962 revealed that cells in the visual cortex are sensitive to small sub-regions of the visual field. Fukushima [17] had proposed neocognitron which is the first model that was simulated on a computer. In [18, 19], LeCun et al. had proposed an algorithm to train the CNN (LeNet-5) using error gradient and compared the performance with other methods on pattern recognition tasks. But at that time due to lack of data and computational power, complex problems such as video classification could not be tackled by LeNet-5. Since the emergence of GPUs and their usage in machine learning [20], the area of CNN has been rejuvenated. AlexNet [21], a deep CNN architecture, was proposed by Krizhevsky et al. which achieved improvement in image classification tasks. AlexNet is similar to LeNet-5, but with a structure that is deeper. With the success of AlexNet various architectures such as GoogleNet [22], VGGNet [23], ZFNet [24], and ResNet [25] were proposed which have shown improved performance, due to all these advancements in CNNs. CNNs also face some difficulty at the time of training because large number of parameters requires heavy computation power, the learning objective is non-convex and has many poor local minima, vanishing/exploding gradients problem and needed a large amount of training data. To overcome the problem of local minima, weight initialization also plays a very crucial role in deep learning.

In this paper, we have proposed a novel method for the initialization of the weights of CNNs using a stacked autoencoder (SAE), in which we learn the weights of the SAE in an unsupervised manner. The number of the hidden layers in the SAE depends on the number of convolutional layers used. The first hidden layer weights act as the first-layer filter weights in the CNN, and second layer CNNs filters are the second layer of SAE weights and so on. In the literature, researches have proposed various method to initialize the weights of CNNs. Glorot and Bengio [26] have proposed Xavier initialization to keep the same variance of each neuron among layers considering that no nonlinearity exists between layers. But Xavier initialization fails when activation units are ReLU type nonlinearity. Basically, ReLU eliminates half of the distribution. He et al. [27] proposed a robust initialization method which is the extended version of the Xavier initialization for the ReLU type nonlinearity in which they consider parametric rectified linear unit (PReLU) that generalizes the traditional ReLU unit. Simonyan and Zisserman in [28] suggested that first trained their most shallow architecture which helps for initializing the second one, and so

forth. Saxe et al. [29] presented a orthonormal matrix initialization which works better for linear networks for Gaussian noise. It also works for the networks with nonlinearities. Ioffe and Szegedy [30] inserted batch normalization to reduce the dependence on initialization, and they also used scale and shift operation to preserve the capacity of the network.

The reminder of the paper is organized as: Sect. 2 deals with basic background of SAE and CNNs, Sect. 3 describes the proposed methodology, Sect. 4 discusses the result and discussion, and finally, Sect. 5 concludes the paper.

2 Basic Background

2.1 Stacked Autoencoder

A stacked autoencoder is a multilayer neural network in which more than one autoencoder (AE) is stacked and trained using the greedy layer wise [1, 31, 32]. An AE learns more abstract representations from the training data in an unsupervised manner. The training of AE is done using backpropagation (BP) to minimize the error difference between the input and the constructed output. The basic architecture and training of the AE are shown in Fig. 1 and Algorithm 1.

2.2 Convolutional Neural Network

A CNN [33] is comprised of multilayer convolutional layers, pooling layers and then followed by multilayer fully connected layers. There are four main parts of a CNN as explained below.

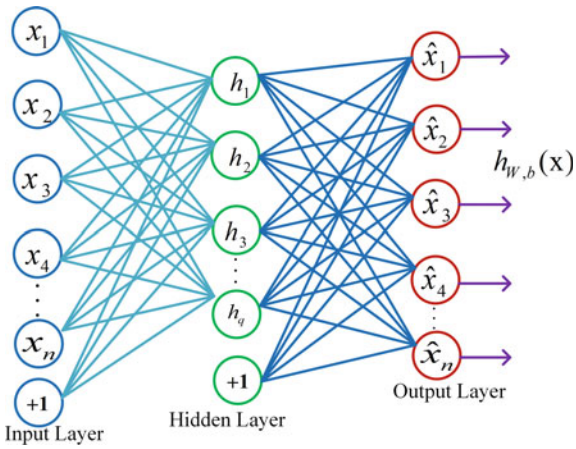


Fig. 1 Basic network of autoencoder [9]

- **Convolution**: Features from the input image are extracted in convolutional layer of the CNN. Using small squares of the input data, image features are learnt by convolution which helps preserve the spatial relationship between the pixels. The output of the convolved image is called the convolved feature or activation map. The size of the convolved feature is controlled by three parameters:

  1. **Depth**: Depth corresponds to the number of kernels (filters) we use for the convolution operation.
  2. **Stride**: Stride is the number of pixels by which the kernel matrix is slid over the input matrix. The filters are moved one pixel at a time when the stride is 1, two pixels at a time when the stride is 2, and so on.
  3. **Zero padding**: Zero padding is used to apply the filter for bordering element of our input image.

- **Nonlinearity**: Nonlinearity in deep networks plays a very important role in learning complicated structures from the data. Various types of nonlinearities are used like Sigmoid, Tanh, ReLU.
- **Pooling**: Pooling reduces the dimension of each feature map but retains the most relevant features. It is also called subsampling or downsampling. It can be of different types: average, sum, max, etc.
- **Fully Connected Layer**: It is basically a multilayer perceptron that uses a softmax function as a activation function in the output layer.

Algorithm 1: Training of autoencoder

Step 1. Initialize vector $\mathbf{b}$, i.e., bias as a zero vector and matrix $\mathbf{W}$ between $[-\nu, \nu]$ randomly, where ν is depends on neurons in hidden layers.
Step 2. Set, $\Delta\mathbf{W}^l$=0 and $\Delta\mathbf{b}^l$=0 for all l
Step 3. Compute gradient using BP algorithm

$$\nabla_{\mathbf{W}^l} J(\mathbf{W}, \mathbf{b}; x, y) = \delta^{l+1}(a^l)^T$$

$$\nabla_{\mathbf{b}^l} J(\mathbf{W}, \mathbf{b}; x, y) = \delta^{l+1}$$

and a. Set $\Delta\mathbf{W}^l := \Delta\mathbf{W}^l + \nabla_{\mathbf{W}^l} J(\mathbf{W}, \mathbf{b}; x, y)$

b. Set $\Delta\mathbf{b}^l := \Delta\mathbf{b}^l + \nabla_{\mathbf{b}^l} J(\mathbf{W}, \mathbf{b}; x, y)$
for all samples of training data. where, a^l is activation of lth hidden layer.
Step 4. Update weight matrix $\mathbf{W}$ and vector $\mathbf{b}$

$$\mathbf{W}^l = \mathbf{W}^l - \eta[\tfrac{1}{m}\Delta(\mathbf{W}^l) + \lambda(\mathbf{W}^l)]$$

$$\mathbf{b}^l = \mathbf{b}^l - \eta[\tfrac{1}{m}\Delta(\mathbf{b}^l)]$$

where, η is learning parameter.

---

3 Proposed Methodology

The main objective is to improve and maintain the performance of a convolutional neural network (CNN) on any given dataset by better initialization of the kernel (filter) weights. In our method, we utilize two networks namely a convolutional neural network and an autoencoder. The network architecture of the autoencoder is chosen such that the number of hidden layers of the autoencoder is equal to the number of convolutional layers in the CNN. Also, it must be noted that the autoencoder network architecture must be chosen in such a way that the number of weights of the autoencoder must be greater than or equal to the total number of weights required for the complete initialization of all the filter weights of the CNN. The first step is data collection, wherein relevant gray-scale images, and the relevant image labels are obtained. Once the data has been properly cleaned, we then go on to the initial steps of training on this dataset. The autoencoder is trained on the collected dataset. An autoencoder tries to create a compressed representation of the image, and then tries to reconstruct the original image as far as possible from that compressed representation. Once the training of the autoencoder is complete, the weights of the autoencoder are stored locally. Hence, essentially, the mapping of the input pixels of the image to that of the compressed representation, and also the mapping from that compressed representation to the original gray-scale image are stored. These results are noted so as to obtain the minimum loss when many hyper-parameters of the autoencoder such as the number of layers, the activation functions of the individual neurons, and the sparsity parameters are all experimented on during the course of training the network.

The CNN is then trained on the same dataset. The main point to be considered now is the initialization method of each of the convolutional layers. The loading of the weights of the trained autoencoder is done such that each of the convolutional layer of the CNN loads the weights randomly from the corresponding layer of the trained autoencoder. The initialization is carried out such that the first layer of the autoencoder contributes to the first convolutional layer of the CNN, the second layer of the autoencoder to the second convolutional layer of the CNN and so on. The initialization is completed, and then the CNN is now trained on the given dataset. The results of the training are noted, after careful experimentation with the hyper-parameters of the network such as the number of filters, the number of layers, and the activation functions of the individual neurons. The CNN with the same network architecture is trained on the same dataset again. Here, in this case, each of the convolutional layers of the CNN are initialized randomly. The method of initialization of the convolutional layers is Xavier initialization. The hyper-parameters of the CNN are kept the same when training with random initialization of weights so as to produce a common ground for better comparisons of this method of initialization with the method involving initialization of the filter weights with those of the trained autoencoder. Finally, the results of both the methods of initialization of the CNN are noted, and comparisons are then made.

4 Results and Discussion

The training method discussed in the previous section was implemented on three datasets, namely MNIST dataset, MIAS MiniMammographic dataset, and CIFAR-10 dataset.

4.1 MNIST Dataset

The MNIST dataset [34] comprises of a total of 70,000 gray-scale images representing handwritten digits, from 0 to 9, provided in a 28 × 28 matrix format, where the pixel intensities are represented as integers having values between 0 and 255, both inclusive. The dataset is split in such a way that there are 60,000 training data (images) and 10,000 testing data (images). The labels are provided as integers from 0 to 9. For the purpose of training the Autoencoder on the dataset, the images are normalized so that all the intensities would lie between 0 and 1, both inclusive.

1. **Autoencoder**: For 70,000 images of size 28 × 28 pixels, the autoencoder consists of 4 layers, namely input layer consisting of 784 neurons, followed by 2 hidden layers, each consisting of 350 neurons, followed by the output layer consisting of 784 neurons. A sparsity of 10^{-7} is maintained in both the hidden layers. All the neurons follow ReLU activation, except the output layer which follows the sigmoid activation function. The data is trained for 150 epochs, with a mini-batch size of 256, following the 'adadelta' optimization.
2. **Convolutional Neural Network**: The CNN works on 70,000 images, each 28 × 28 pixels, falling in 1 of 10 classes. The CNN consists of 2 convolutional layers, each of which is followed by a max pooling layer, finally ending in a fully connected layer. The number of filters in the first layer is 6, and in the second layer is 16, each of which is 5 × 5 in size. The pooling regions are 2 × 2 in size. All the neurons follow ReLU activation, except the final layer which follows the softmax activation function. The network is trained for 20 epochs, with a mini-batch size of 256, following the 'adam' optimization.

The experiment was conducted in a normal laptop with a NVIDIA GeForce 920M GPU. The language used is Python version 2.7.13 with the Keras library version 2.0.6 running on backend of Theano with version 0.9.0, installed in a conda virtualenv with the anaconda version as 4.4.0 running on Ubuntu 16.04. For the CNN initialized with Xaviers method of initialization of weights, the total run time taken for the training was 2 min, 48 s, whereas the time per epoch was 7 s. The network achieved 99.115% accuracy on the training set, and 98.93% on the test set as shown in the Table 1. The autoencoder took 7 min, 50 s to train and took 3 s per epoch. The initial training loss was 0.3240, and the initial test set loss was 0.2622. After training, the final training loss was 0.0850, and the final test set loss was 0.0845. The CNN initialized with the autoencoder weights took a total time of 2 min and 39 s to train. The time per epoch

was 6 s. The training set accuracy was 99.00% and the test set accuracy was 98.91% as shown in the Table 1.

4.2 CIFAR-10 Dataset

The CIFAR-10 [35] dataset consists of a total of 60,000 RGB images, each of which represents an object from any one of the following classes, airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The images are provided in a 32×32 pixels format. The dataset is split in such a way that there are 50,000 training images and 10,000 test images. The labels are given as integers between 0 and 9. To maintain the same network architecture across the 3 datasets, the RGB images are converted into gray scale. For the purpose of training the autoencoder on the dataset, the images are normalized so that all the intensities would lie between 0 and 1, both inclusive.

1. **Autoencoder**: For 60,000 images of size 32×32 pixels, the autoencoder consists of 4 layers, namely the input layer consisting of 1024 neurons, followed by 2 hidden layers, each consisting of 400 neurons, followed by the output layer consisting of 1024 neurons. A sparsity of 10^{-7} is maintained in both the hidden layers. All the neurons follow ReLU activation, except the final layer which follows the sigmoid activation function. The data is trained for 150 epochs, with a mini-batch size of 256, following the 'adadelta' optimization.
2. **Convolutional Neural Network**: The CNN works on 60,000 images, each 32×32 pixels, falling in 1 of 10 classes. The CNN consists of 2 Convolutional layers, each of which is followed by a MaxPooling layer, finally ending in a fully connected layer. The number of filters in the first layer is 6, and in the second layer is 16, each of which are 5×5 in size. The pooling regions are 2×2 in size. All the neurons follow ReLU activation, except the final layer which follows the softmax activation function. The network is trained for 250 epochs, with a mini-batch size of 256, following the 'adam' optimization.

The CNN initialized with Xaviers method of initialization of weights, the total time taken for the training was 24 min, 36 s, whereas the time per epoch was 7 s. The network achieved 66.926% accuracy on the training set, and 62.49% on the test set as shown in Table 1. The autoencoder took 7 min, 37 s to train, and took 3 s per epoch. The initial training loss was 0.6352, and the initial test set loss was 0.6309. After training, the final training loss was 0.5782, and the final test set loss was 0.5793. The CNN initialized with the autoencoder weights took a total time of 24 min and 11 s to train. The time per epoch was 7 s. The training set accuracy was 63.694% and the test set accuracy was 59.27% shown in Table 1. From the above results, we observe that the accuracy of the CNN that has been initialized with the trained autoencoder weights randomly has a performance which is comparable to that of the CNN which has its weights initialized randomly using Xavier initialization. The accuracy is a bit on the lower side as the spatial connections provided by the CNN is not learnt and preserved by the autoencoder, and hence, for a dataset such as CIFAR-10

Table 1 Performance of various learning in the term of test accuracy (%)

Datasets	Xavier initialization		Proposed initialization	
	Training accuracy	Testing accuracy	Training accuracy	Testing accuracy
MNIST [34]	99.12	98.93	99.00	98.91
MIAS MiniMammographic [36]	63.40	69.57	63.40	69.57
CIFAR-10 [35]	66.93	62.49	63.69	59.27

which consists of natural images, the accuracy of the CNN whose weights have been initialized with that of the trained autoencoder is lower. Hence, the method of initialization of CNN weights randomly from those of the autoencoder weights can be used as an alternate method of initialization since the performance is comparable between the two methods.

4.3 MIAS MiniMammographic dataset

The MiniMammographic dataset [36] consists of 322 gray scale images, each of size 1024 × 1024 pixels, which are given in pgm format. There are 7 classes of the abnormality present, which are as follows: CALC, CIRC, SPIC, MISC, ARCH, ASYM, and NORM. These labels are converted from the textual representation to integers between 0 and 6 for the purpose of training. The data is randomly split such that 14% of the images are used for testing. For the purpose of training the autoencoder on the dataset, the images are normalized so that all the intensities would lie between 0 and 1, both inclusive.

1. **Autoencoder**: For 322 images of size 1024 × 1024 pixels, the autoencoder consists of 4 layers, namely the input layer consisting of 1,048,576 neurons, followed by 2 hidden layers, each consisting of 100 neurons, followed by the output layer consisting of 1,048,576 neurons. A sparsity of 10^{-7} is maintained in both the hidden layers. All the neurons follow ReLU activation, except the final layer which follows the sigmoid activation function. The data is trained for 3 epochs, with a mini-batch size of 16, following the 'adadelta' optimization.
2. **Convolutional Neural Network**: The CNN works on 322 images, each 1024 × 1024 pixels, falling in 1 of 7 classes. The CNN consists of 2 Convolutional layers, each of which is followed by a max pooling layer, finally ending in a fully connected layer. The number of filters in the first layer is 6, and in the second layer is 16, each of which is 5 × 5 in size. The pooling regions are 2 × 2 in size. All the neurons follow ReLU activation, except the final layer which follows the softmax activation function. The network is trained for 3 epochs, with a mini-batch size of 32, following the 'adam' optimization.

The CNN was trained on the dataset by using the fivefold cross-validation technique, with 3 epochs per fold. For the CNN initialized with Xaviers method of initialization of weights, the time per epoch was 115 s. The network achieved an average of 63.404% accuracy on the training set, 69.57% on the test set, and 63.40% on the validation set over fivefolds of training. The autoencoder took 120 s per epoch. The initial training loss was 0.6932, and the initial test set loss was 0.6931. After training, the final training loss was 0.6931, and the final test set loss was 0.6931. The CNN initialized with the autoencoder weights took 173 s per epoch. The training set accuracy was 63.404%, the test set accuracy was 69.57%, and the validation set accuracy was 63.40% over 5 folds of training.

From the above result shown in Table 1, we find that for the MNIST dataset and for the MIAS MiniMammographic dataset, the initialization with the autoencoder weights makes the performance of the CNN comparable to that of Xavier initialization. The results of those of the CIFAR-10 dataset are slightly lower than that of Xavier initialization because the autoencoder might not have been able to capture all the spatial features, and connections present in everyday color images that have been converted into gray scale. Hence, from the above table of results, we can conclude that the accuracy of the CNN which has been initialized with the weights of the autoencoder are comparable to that of the CNN which has been initialized using Xavier initialization method.

5 Conclusion

As we know, a CNN is a non-convex optimization problem. In the case of a non-convex optimization problem, the optimal solution depends on the initial parameters. In the most of the literatures, authors have used various methods for initialization. In the paper, the initial parameters of the CNN are initialized using the SAE. By initialization using the SAE weights, the performance of the CNN is also comparable to Xavier initialization as shown in Table 1. To validate our network, we have also done fivefold cross-validation on the MiniMammographic dataset.

References

1. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
2. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, pp. 153–160 (2007)
3. Poultney, C., Chopra, S., Cun, Y.L.: Efficient learning of sparse representations with an energy-based model. In: Advances in Neural Information Processing Systems, pp. 1137–1144 (2007)
4. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 3371–3408 (2010)

5. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, pp. 37–49 (2012)
6. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation (No. ICS-8506). California University, San Diego, La Jolla, Institute for Cognitive Science (1985)
7. Sevakula, R.K., Thirukovalluru, R., Verma, N.K., Cui, Y.: Deep neural networks for transcriptome based cancer classification. BMC Bioinform. (2017) (Accepted)
8. Rajurkar, S., Singh, V., Verma, N.K., Cui, Y.: Deep stacked auto-encoder with deep fuzzy network for transcriptome based tumor type classification. BMC Bioinform. (2017) (Accepted)
9. Singh, V., Verma, N.K.: Deep learning architecture for high-level feature generation using stacked auto encoder for business intelligence. In: Complex Systems: Solutions and Challenges in Economics, Management and Engineering. Springer International Publishing (Accepted) (2017)
10. Sevakula, R.K., Singh, V., Verma, N.K., Kumar, C., Cui, Y.: Transfer learning for molecular cancer classification using deep neural networks. IEEE/ACM Trans. Comput. Biol. Bioinform. (1), 1–1 (2018)
11. Singh, V., Gupta, R.K., Sevakula, R.K., Verma, N.K.: Comparative analysis of Gaussian mixture model, logistic regression and random forest for big data classification using map reduce. In: 2016 11th IEEE International Conference on Industrial and Information Systems (ICIIS), pp. 333–338 (2016)
12. Verma, N.K., Sharma, T., Rajurkar, S.D., Salour, A.: Object identification for inventory management using convolutional neural network. In: 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–6 (2016)
13. Sevakula, R.K., Verma, N.K.: Assessing generalization ability of majority vote point classifiers. IEEE Trans. Neural Netw. Learn. Syst. **28**(12), 2985–2997 (2017)
14. Rajurkar, S., Verma, N.K.: Developing deep fuzzy network with Takagi Sugeno fuzzy inference system. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6 (2017)
15. Verma, N. K., Singh, S.: Image sequence prediction using ANN and RBFNN. Int. J. Image Graph. **13**(02), 1340006, (2013)
16. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. **160**(1), 106–154 (1962)
17. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**(4), 193–202 (1980)
18. Le Cun, Y., Boser, B., et al.: Handwritten digit recognition with a back propagation network. Adv. Neural Inf. Process. Syst. (1990)
19. LeCun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
20. Steinkrau, D., Simard, P.Y., Buck, I.: Using GPUs for machine learning algorithms. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition. IEEE Computer Society (2005)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. (2012)
22. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
24. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Springer International Publishing (2014)
25. He, K., et al.: Deep residual learning for image recognition. arXiv:1512.03385 (2015)
26. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)

27. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034 (2015)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
29. Saxe, A.M., McClelland, J.L., Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120 (2013)
30. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167 [cs] (2015)
31. Ng, A., Ngiam, J., Foo, C.Y., Mai, Y., Suen, C.: UFLDL Tutorial (2016)
32. Singh, V., Baranwal, N., Sevakula, R.K., Verma, N.K., Cui, Y.: Layerwise feature selection in stacked sparse auto-encoder for tumor type prediction. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1542–1548. IEEE (2016)
33. http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/
34. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010). http://yann.lecun.com/exdb/mnist/
35. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images (2009)
36. Suckling, J., et al.: The mammographic image analysis society digital mammogram database. In: International Congress Series, vol. 1069, pp. 375–378 (1994)

Behavioural Analysis of Recent Ransomwares and Prediction of Future Attacks by Polymorphic and Metamorphic Ransomware

Navneet Kaur Popli and Anup Girdhar

Abstract Attackers understand that data, files, networks and all digital resources are a key factor in the regular working and growth of any business. And because these digital resources are so dear to the business, the best and quickest way to earn big money is to hold all these resources at ransom. Thus, born ransomware are malwares that normally encrypt all your important files and ask for money to provide you with the decryption key. Some ransomware enters the system through social engineering, while others try to find vulnerabilities, open ports or employ backdoors to get inside. In any case, their main aim is commercial gain. Ransomware is the biggest threat to enterprises and has the potential to bring business at standstill and cause huge loss to economy. In this paper, we do an analytical study of the behaviours of the recent ransomwares that have attacked businesses and individuals alike. We run them in a simulated environment and analyse their attack process, file system analysis, persistence analysis and network-level analysis. We also employ tools like Cuckoo to do this behaviour analysis part. After that, we predict the future types of ransomwares which can be created easily by using available toolkits like ADMMutate, Clet and Phatbot. We also predict the impact and the threat they can cause and how difficult it would be to detect them after they employ all the mentioned stealth techniques.

Keywords Ransomware · IDS · Encryption · Shellcodes · Polymorphism Metamorphism · Cybercrime

N. K. Popli (✉)
Tilak Maharashtra Vidyapeeth, Pune, India
e-mail: navneet_popli@yahoo.co.in; navneetkaurpopli@gmail.com

A. Girdhar
Sedulity Solutions and Technologies, Delhi, India
e-mail: anup@sedulitygroups.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_6

1 Introduction

Ransomwares are a big danger to the normal working of business organisations. When an attack by the ransomware happens, it can stall the biggest enterprises and can bring the business and the economy to a halt. The paper is divided into four sections. The first section studies the impact that the latest families of ransomwares have affected organisations, businesses and individuals. The second section describes the behavioural analysis of different kinds of ransomware, which recently attacked businesses: WannaCry and Petya. We understand their attack procedure, file system analysis, persistence and network indicators' perspective. The third section talks about the defence mechanisms and the detection and mitigation policies to be followed by IDS' to protect the systems from these attacks. The fourth section envisions the kind of ransomwares that can be created in the future and the damages that they can make and how the existing IDS' may fail to detect them. Finally, we conclude by suggesting some new policies which should get integrated with IDS' to conquer these attacks.

2 Literature Review

Ahmadian et al. [1] propose a taxonomy of various kinds of ransomwares, and then based on some common features, they propose a framework for their detection at an early stage so as to prevent them from encrypting the user data. They experimentally prove its effectiveness as well. Gazet [2] technically reviews and compares many ransomwares based on quality of code, malwares' functionalities and analysis of cryptographic primitives if any. Based on this analysis, he tries to find out the strength and weakness of these extortion means, their business model and the communication going around them. Weckstén and Frick [3] propose a novel method for recovery from crypto-ransomwares. They analyse four types of ransomwares and provide a solution in the form of a script which renames the tool which handles shadow copies in order to recover from infections. Tseng et al. [4] study different signatures or behaviours of different ransomware families and propose a deep learning solution for their detection. Lozada [5] analyses the impact of ransomware on health care and economy. He also talks about the steps required to mitigate the risks. He not only discusses about the various infiltration methods for ransomware, but also tells about the growth of ransomware over the years and the impact they have on the economy. Ganorkar and Kandasamy [6] does a static analysis to differentiate between different types of ransomware. He also emphasises the role cryptocurrency plays in a ransomware attack and how the fact that all transactions are completely anonymous strengthens the attack by ransomware. Mercaldo et al. [7] manually dissects various ransomware samples and finds out a number of static and behavioural features of these ransomware. Then, he tries to analyse which of these features are representative of general ransomware. Cabaj et al. [8] discusses the network behaviour of Cryptowall

ransomware sample. He uses Honeypot infrastructure to analyse the network activity and get understanding of the dynamic behaviour in a dedicated environment. Ferrante et al. [9] tries to follow static and dynamic detection methods for mobile ransomware. After analysing the ransomware behaviour, he compares the detection methods to find out which is a better detection technique. In the process, he understands the entire activity of several ransomware from attack to encryption stage.

3 Idea of Research

Ransomware attack can impact businesses and can adversely affect economies. In the recent ransomware attack like those of WannaCry, they can be a threat to basic public services like health care, food delivery, transportation and even communication services. The reason for attack is mostly commercial, however, as in the case of attacks like that of Petya; the reason may be destruction of data as well. In some cases, a victim may be able to recover data after paying ransom, while in others there is no guarantee of getting the data back even after paying the bitcoins. Massive research has been going on for the detection and mitigation of these attacks. However, researchers are one step behind when attackers try to employ obfuscation techniques or try to convert the ransomware into polymorphic or metamorphic code. Then, it becomes almost impossible to detect and mitigate it. In this paper, we have first analysed the attack process and the behaviour of recent ransomware. Thus, we understand how they work. The main focus however is to find out what techniques can be applied by the ransomware to get converted into polymorphic and metamorphic malwares. These future strains of malware need advance techniques of detection and mitigation. We have also tried to predict what havoc can these type of future ransomware can cause and what we can do to protect ourselves from them.

4 Recent Impact of Ransomwares

4.1 WannaCry

Friday, May 12, 2017 was the day when about 40 hospitals of National Health Service in UK had to shut down their operations because of a major attack on their computer systems by a ransomware called 'WannaCry'. The entire data in these systems was encrypted, and lives of many patients were in danger. Similar was the case of 'FedEx' in US, 'Telefonica' in Spain, 'Deutsche Bahn' in Germany and 'Latam Airlines' in South America. Ultimately 150 countries with 2,00,000 computers were attacked by WannaCry, and by Monday, some $50,000 were already paid to the attackers by various companies and individuals. However, not everyone's data was recovered

even after the ransom was paid. The reason for that is one cannot tie payment to who you are making it to [10].

4.2 Petya

Petya attacked in June 2017, just one month after WannaCry using almost the same SMB vulnerability exploit and NSA's EternalBlue. However, since many systems were already patched, its impact was very limited. The bitcoin payment system was very simplified and traceable, and not much money was earned in bitcoins. This makes one ponder about whether the real purpose of the attack was actually monetary profit? Petya's first target was the Ukrainian systems, both local and government. It entered the systems through a poisoned update for MeDoc accounting software. MeDoc is primarily used in Ukraine and Russia. It affected Ukrainian banks, airports, power companies and even Chernobyl Nuclear Plant. Outside Ukraine, it affected Danish shipping company 'Maersk', French company 'Saint Gobain', British advertising company 'WPP', Russian oil production company 'Rosneft' and even 'Jawaharlal Nehru port' in Mumbai, India. Till June 28, less than 150 organisations in Ukraine and less than 50 in US were affected. Thus, Petya's main focus was not a fast lateral spread or financial gain but to destroy data, especially in Ukraine systems.

5 Reasons of Attack

There are many reasons why ransomwares like WannaCry and Petya are successful in attacking the systems. Some of the reasons are as follows:

5.1 Entering by Social Engineering

These ransomwares are normally delivered via e-mails which will lure the recipient to click on malicious attachments to open them. This will release the ransomware on the machine, and then, it would start spreading across the network after infecting the first machine. The ransomware can also enter the machine by downloading a bad app or getting attachments from malicious sites.

5.2 Not Updating Systems Regularly

All operating systems regularly release patches for their updation after analysing different security threats. In the case of WannaCry, Microsoft had already released

a patch called MS17-010 after realising about the NSA leak. However, many people failed to update their machines with this patch and were thus vulnerable to ransomware attack. Three years ago, Microsoft stopped supporting Windows XP systems. Thus, those were the firsts to be attacked. A patch was released for them also after the attack.

5.3 SMB Vulnerability

In MS Windows, there are some ports which are open by default [11]. One of them is the Service Message Block (SMB) port used for file and printer sharing. WannaCry and Petya both spread using this open port.

5.4 No Network Segmentation

If the network is not properly segmented, the ransomware spreads at a very fast rate. However, if the network is segmented, then the spread of the ransomware can be contained and the mitigation may also be easily possible.

5.5 Good AV or IDS Solution

A good and reliable antivirus and IDS solution goes a long way in protecting the network from known and unknown attacks. Antiviruses keep updating themselves when new strains of malwares are discovered. Most of the malwares are successfully blocked by good antiviruses.

5.6 Backup and Restore

Always take a regular backup of your data so that even if the ransomware attacks, you have the most current backup and are able to restore your data without paying the ransom amount and becoming a victim to the attacker.

5.7 Make a Vulnerability Public

As in the case of EternalBlue, the SMB vulnerability was known to NSA, but they chose not to disclose it to Microsoft; rather, they made cyber weapons for them. These

cyber weapons were leaked by Shadow Brokers group, and WannaCry and Petya were created out of them. It is always in the interest of the public and government both that any vulnerability in any operating system, if known by an individual or an organisation, should be made public so that systems can be protected from it.

6 Behavioural Analysis

6.1 WannaCry

6.1.1 Attack Process

Once WannaCry enters the victim computer, it scans heavily for the SMB port vulnerability. If the port is found open, an SMB connection is set up. Then, the patch MS17-010 is searched for. If it is not found then an encrypted shellcode is prepared in Base64 encoding and is heap sprayed in the memory. Once found, the shellcode looks for DoublePulsar. This backdoor changes permissions to remote access and calls a particular domain to see if it is registered. This is done to find out whether the malware is working in a sandbox or not. If the domain is not registered, SMB exploitation starts getting performed. A file called 'taskche.exe' is created which will carry out file encryption and will also help the malware spread across the network. In the meantime, TOR is used for all communications, bitcoin wallets are loaded, and public and encryption key is prepared. The files start getting encrypted. WannaDecryptor.exe is set up, and Please_Read_Me.txt file is created. The screen is locked, and ransom message is displayed.

6.1.2 File System Analysis

The payload is 'taskche.exe' which is created from the worm's resource '1831'. It has a huge size because of bundled TOR executables along with other tools and configuration files. The actual name of this file is 'DiskPart.exe'. This is a MS utility for disc partitioning. WannaCry creates two DLL files in the memory—a 32-bit dll and a 64-bit dll. Both of them contain an export called 'PlayGame'. This writes a copy of the original worm to C:\WINDOWS\mssecsvc.exe and executes it. There are many files which are created in the process. B.wnry is a bitmap image containing a ransom note in it. C.wncr is a binary configuration file. It also has addresses of the TOR sites. R.wnry is a text file with a ransom note in it. S.wnry is a zip file with TOR executable. T.wnry is a dll. It is actually an encryption tool. Taskse.exe starts @WannaDecrytor@.exe. It is a support tool. U.wnry is a decryptor executable that opens GUI with a ransom note in it. Taskdl.exe is a support tool for deleting temporary files. msg\m_*.wnry is a directory with ransom note in different languages like English, Chinese, Bulgarian. If WannaCry is launched *i*th less than 2 arguments,

it installs a service called mssecsvc.exe. This is Microsoft Security Center Service, version 2.0. It drops the WannaCry binary and runs it. If the malware was run with two or more arguments, it enters the service mode. Files are encrypted with the extension of. wncrypt. Some libraries are loaded at run-time like kernel32, user32, advapi32.dll, shell32.dll, msvcrt.dll, mscp60.dll.

6.1.3 Persistence Analysis

Persistence ensures that the malware will continue to run even after the machine is rebooted, restarted or is logged off. For persistence, entering into the run keys in the registry is important. Two registry entries are required to ensure persistence:

- Key:HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\<Random> Value: <Full_path>\tasksche.exe. CU stands for Current User. %Random is a pseudorandom name derived from the current computer name. This is a user-level entry.
- Key:HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\<Random> Value: <Full_path>\tasksche.exe. LM stands for local machine. This is a system-level entry. Full_path is the path of the exe file.

If the DiskPart.exe file finds that it was running without the '/i' switch, this means that the worm has not executed it. Therefore, the file registers itself as a service so that it gains persistence so that the worm is no longer required to run it.

6.1.4 Network Analysis

The network analysis is obtained by observing the pcap file. The important property of this ransomware which makes it different from the previous ransomwares is that it does not need to call a malicious server for encryption of the files. If the binary is in the system, it does not need to be connected to the Internet for encryption of the files. Other ransomwares before it used to connect to the c&c servers to get the encryption key. WannaCry starts its execution by trying to connect to a domain 'www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com'. If the machine fails in making the connection, the exploitation continues. However, if the connection is successful, the exploitation is stopped immediately. Actually, this is a way of 'Sandbox Evasion' technique. Once WannaCry is sure that the domain is unregistered, the exploitation begins. However, it searches for a backdoor called 'DoublePulsar' before actual exploitation. So the attacker sends a SMB 'trans2SESSION_SETUP' request to the infected system. This is a Transaction 2 Subcommand Extension. This is to find out whether the system is already compromised with DoublePulsar or not. The system responds with a 'Not Implemented' message, and a 'Multiplex ID' is returned as part of this message. This has a value of 65 (0×41) for the normal systems and 81 (0×51) for infected systems. If the system is infected, then SMB can be used as a covert channel to install WannaCry. Then, the exploitation starts by setting up Windows Socket APIs. WannaCry uses MS17-010 exploit to laterally spread to other machines

through open NetBIOS ports [12]. Once it finds a machine with an open NetBIOS port, it sends three NetBIOS session setup packets to it. One contains IP address of the machine to be exploited, and the other two contain two IP addresses which are hardcoded in the malware only. It then spawns 2 threads: the first thread enumerates the network adapters and finds the address of the subnet of the compromised system. The first thread then generates threads for each IP address on this subnet. Each of these threads tries to get connected to port 445, and if successful, the entire process of exploitation starts again on the newly infected machine. One hundred and twenty-eight instances of the second thread can be created within 2 s. This explains the widespread of this malware within a short span of time. WannaCry also downloads files from the dist.torproject.org, and all further communications for bitcoin transfer happen through TOR only.

6.2 Petya

6.2.1 Attack Process

Petya was a ransomware which came in March 2016. The strain of this ransomware which came on June 27, 2017 was very different from the original version. Thus, it was called 'NotPetya'. The ransomware uses EternalBlue exploit. The infection starts with a poisoned update for the MeDoc accounting software [13]. Petya infects the MBR and executes a malicious payload that encrypts the file system of the hard drive and does not let Windows system to boot. Like WannaCry takes the help of DoublePulsar for completing its execution, in a similar manner, Petya has joined hands with a traditional file-based ransomware called 'Misha'. Petya needs to have administrative privileges to carry out low-level encryption of files and other malicious works. If those privileges cannot be obtained, Misha is launched. A deep analysis of Petya shows it is actually a 'Wiper'. Normally files encrypted by Petya cannot be recovered. It wipes away the files, and they can never be found again.

6.2.2 File System Analysis

Petya's infection mechanism is to attack low-level structures. It overwrites the beginning of the disc which is the master boot record (MBR) by its own boot loader. This boot loader then loads its 32 sector kernel which then starts the encryption process of the files. AES-128 is used for encryption, and the 128-bit key is randomly generated. This key is again encrypted with RSA 20148 encryption algorithm. Petya starts with the file explorer.exe. Then, ezvit.exe affects the MeDoc software. Then, rundll32.exe is dropped and unicryptC.exe is used for encryption.

6.2.3 Persistence Analysis

MBR and NTFS boot record are the only persistence mechanisms in case of Petya.

6.2.4 Network Analysis

For lateral propagation, in addition to EternalBlue exploit, WMI Commands, Mimikatz and PSExec were also used. Unlike WannaCry which infected a small number of computers and was designed to spread at a fast rate to the Internet, Petya is designed mostly for lateral propagation in the local network. It infected a large number of computers, but the spread was slow and local rather than global. The SMB ports were scanned. If they were found to be unpatched for the EternalBlue exploit, this vulnerability was used to compromise the systems. However, if a patch was found, Windows Management Instrumentation Command (WMIC) was used for stealing local machine credentials using Mimikatz and then finding remote shares to spread. The WMI is a complete infrastructure for the Windows operating system which does management and administration of the operating system both locally and remotely. It is a very powerful tool, but the same power can be used by attackers for remotely executing ransomware in target machines. There are various WMIC commands which were used by Petya for several purposes like detection of antivirus and virtual machines, persistence, theft of data, lateral movement and execution of various commands [14]. Mimikatz is used for stealing user credentials and through them executing Petya on infected machines. One of the WMIC commands used by Petya for executing powershell on a remote machine is:

wmic/node: [IP Address]/user: "[user name]"/password: "[password]" process call create "cmd/c powershell.exe user".

7 Creating Future Strains of Advance Ransomwares

CryptoLocker and WannaCry ransomware did not use any anti-thwarting analysis measures. No obfuscation was used. Petya uses a normal XOR command for obfuscation. Even without or with very little obfuscation also, these ransomwares were able to cause so much damage. Imagine a situation where all different kinds of obfuscation techniques along with blending attacks and techniques for changing the behaviour of code at run-time are applied on a malware. Combine it with social engineering and using different vulnerabilities, and you get a polymorphic or metamorphic ransomware which is non-conquerable. Such a ransomware may cause havoc in the world. Therefore, it is imperative that all IDS and AV developers must know the internal workings of these types of ransomwares to protect their organisations from these new strands of ransomware. The base for all these ransomwares are tiny shellcodes which are embedded deep inside the ransomware body. So applying various

polymorphic and metamorphic techniques on these shellcodes, we can practically do anything with these ransomwares other than just encrypting the files.

Steps in the creation and execution of polymorphic and metamorphic ransomwares:

1. Obfuscation of NOP sled
2. Obfuscation of the rest of the code
3. Ability to launch a multi-staged attack
4. Sandbox evasion techniques
5. Polymorphic blending
6. Conversion to metamorphic code.

7.1 Obfuscation of NOP Sled

Normally, the buffer overflow attack follows very simple steps. First, the attacker finds a vulnerable program, which is most often a program with a buffer (an array) whose bounds are not being checked by the program. In a typical memory layout of this program, the used buffer will be followed by the Extended Instruction Pointer (EIP), which is the address of the program counter or the return address that the execution would jump to when the current function has finished. The attacker would overflow this buffer with a crafted input so that it overwrites the buffer and the EIP and instead points to the shellcode which the attacker wants to execute. This is a precise address where the shellcode is located. Now, the operating system designers are well aware of the buffer overflow attacks. Therefore, to get over these attacks, they have devised a process called 'Address Space Layout Randomisation'. In this process, the address spaces including stacks, heaps and other memory structures are randomly offset. This makes guessing of the accurate location of the shellcode all the more difficult. To get over this problem, the attackers increase the address space using NOPs. NOP stands for no operation. These are one-byte instructions [4] which actually perform no operation but take space in the instruction stream, but do not affect any programmer-accessible registers, flags or memory. It only affects the EIP and just takes the control flow to the next instruction without affecting the program in any way. A series of consecutive NOP instructions is called a NOP sled. The NOP sled is appended before the shellcode which the attacker wants to execute. This ensures that the CPU jumps somewhere in the sled to ultimately reach the shellcode to start the attack.

This boon of NOP sleds can easily become a bane for the attacker. The moment a series of NOP instructions are visible to the antivirus or the IDS, they will block the shellcode from executing. Here, we come to a technique called obfuscation where we replace the NOPs with some other equivalent instructions to bypass the AVs and the IDS.

There are multiple ways in which NOP sleds can be obfuscated [15]:

1. Single-byte NOP equivalent instructions
2. Multiple-byte NOP equivalent instructions
3. Four-byte aligned sled
4. Trampoline sled.

7.2 Obfuscation of the Rest of the Code

In addition to the NOP sleds, the rest of the code can also be obfuscated using various techniques like [16] dead code insertion, subroutine and instruction reordering, code transposition, instruction substitution, code integration and register reassignment. With code obfuscation, other techniques which can be employed are entry point obfuscation, information exfiltration obfuscation, obfuscation of communication to the c&c server, avoiding a big return address and hiding it and self-modification of code.

7.3 Ability to Launch a Multi-staged Attack

Malwares should never be launched in one go. There are various stages of a cyber attack like information gathering, entering the target system, privilege escalation, establishing a connection back to the attacker, injecting a c&c component into the target, going deeper into the network and finally cleaning up the mess, i.e. clearing all signs of the attack so that the malware remains hidden from the antivirus or IDPS software.

7.4 Sandbox Evasion

A very powerful technique developed by antivirus and IDPS creators is sandbox technique. Using this technique, the suspicious programs can be run in a simulated environment called a sandbox. All programs, whether malicious or non-malicious, would show their true nature in this environment. Thus, the anti-malware program would be easily able to detect the malicious programs and weed them out of the system [13]. However, attackers have come up with sandbox evasion techniques. They employ various methods to detect whether they are working in a simulated environment or the actual operating system. Accordingly, if they detect a sandbox, they would lie dormant or do very superfluous jobs. It's only when they are sure they are running in the actual operating system, would they show their true working and would launch an appropriate attack.

7.5 Polymorphic Blending

Anomaly-based IDPS, especially those that are byte frequency-based, finds out about the presence of malware in the network stream by checking the payload behaviour anomalies of the malware. The indicators they track are input and output bytes, number of connections, number of packets, type of protocols used and the communication destination. As an example, data exfiltration can be a powerful network statistic to detect a malware. Say a system is baseline in the sense that it transfers data only to the internal systems. If a large amount of data starts going out of the system to an external server on the Internet, this change in behaviour can raise an alarm and alert the detection system. Also, increase in the packet frequency may indicate a denial of service attack. For detecting host-level attacks, we detect protocols in the packets to see if they have ambiguous options, are too small or violate some application layer protocols. Attackers can evade anomaly-based detection systems by blending their traffic with normal traffic. They prepare the payloads and the behaviours of their packets in such a manner that the packets are indistinguishable from normal traffic. The message HTTP headers can be spoofed to merge with legitimate traffic. For example instead of the name of a c&c server, after connection, the host name can be given as google.com. Also TOR traffic can be disguised to look like a normal HTTPS traffic. Such blending techniques can be used to easily blend malware with normal traffic and evade anomaly-based IDPS.

7.6 Conversion to Metamorphic Code

The principle difference between polymorphism and metamorphism is that polymorphism does not change the original code. It just hides it. However, metamorphism morphs or changes the code body itself. In other words, metamorphic code is body polymorphic. Various metamorphic engines can be applied to convert polymorphic malware into metamorphic malware and make it all the more difficult to detect.

Algorithm of the proposed scheme of creating future strains of ransomware:

```
Begin
creates a legitimate host program 'LHost.exe'.
Create a vulnerability in it 'vul'.
Inject a shellcode 'shellgQ.gen' in LHost.exe.
Insert NOP sled 0X90 in LHost.exe.
while (EIP!=addr(shellQ.gen))
{
        EIP++
}
Obfuscate 0X90.
Obfuscate rest of the code of LHost.exe==LHost1.exe.
Multi Staged attack by LHost1.exe on target system
Convert LHost1.exe==metamorphic(LHost1.exe)
End
```

8 Predicted Threats by Future Ransomware and Malware

After employing all the above-mentioned techniques, it would become very difficult for the IDPS to detect and mitigate these future malwares and ransomwares. WannaCry and Petya whose behaviour was analysed were recent ransomwares which caused huge impact to businesses. WannaCry did not use any obfuscation technique, Petya had an obfuscated dropper, but the rest of the techniques were not implemented. Ransomwares before WannaCry like CryptoLocker, Locky, zCrypt, Cerber and Cryptowall caused huge damage, but none of them used the techniques described above. Employing these techniques is not very difficult because of the presence of many easily available engines which provide these services in quite a user-friendly manner. These polymorphic and metamorphic ransomwares can easily evade both signature-based and anomaly-based IDPS systems [17]. The damage they can cause can be immeasurable. Using either social engineering or some existing vulnerability, they just have to enter one system and then laterally spread in the entire network and can even spread to different networks through Internet, all of it without the fear of any kind of detection.

There are a number of predictions which can be done for the future ransomwares after application of the proposed method, and we can understand how these ransomwares will then be impacting business, government and people in general:

8.1 Beyond Pattern Matching

Signature detection using pattern matching will no longer be possible if ransomwares are obfuscated using the proposed method. Thus, even advance signature detection fails if the ransomware is polymorphic in nature.

8.2 Polymorphic Blending of Traffic

The future ransomware will be able to blend their traffic with normal traffic using this method. Thus, all malicious traffic would seem like normal traffic and regular monitoring of network traffic for detection of malicious activity would also fail.

8.3 Sandbox Evasion Techniques

Since the ransomware created using our method also employs sandbox evasion techniques, dynamic behaviour analysis in a simulated environment would fail in this case. As long as the ransomware knows it is working in a sandbox, it would remain dormant without showing any malicious activity and would thus seem perfectly legitimate.

8.4 Multi-staged Attack

Using the proposed method, the attacker follows a very organised routine by gathering complete data about the target machine, escalating privileges and lateral spread. In addition, the ransomware also cleans the memory of all its traces of malicious activity so that it bypasses all antiviruses and keeps on staying in the system.

8.5 IOT Systems at Heavy Risk

Since technology is pervading all spheres of human life, wearable computers are becoming increasingly popular among masses. The dependence of people on these devices is growing. Smart cities are mushrooming everywhere, and people are thronging to live in them. With increasing demand, the risk is also increasing [18]. Not only organisations but more and more individuals can be affected by advance ransomware.

8.6 Public Services Can Get Stalled

All basic public services like health care, transportation, water and electricity are completely dependent on technology. If ransomware attacks them, basic public amenities would be hugely affected and life may come to a standstill. If these essential services are halted or this information corrupted or wiped off, it would cause a huge loss to governments as well as cause a lot of inconvenience to the general public.

8.7 Enterprise Ransomware

In the future, ransomware will not attack a random system and then try to spread laterally; rather, they would attack the enterprise server itself. Asking for a million dollars is much more efficient rather than asking for a few hundred dollars for a single computer.

9 Conclusion

Polymorphic and metamorphic ransomware can apply the above-mentioned techniques and can become almost untraceable. They can then create havoc not only in the existing network but in all other networks through the Internet. To stop these

future malwares, IDPS has to update their policies so that they are able to detect these ransomwares. Advanced signature matching can be employed to detect polymorphic signatures, and emulation-based detection systems can detect these malwares based on their behaviour in a simulated environment. Anomaly-based detection systems should be made more efficient by employing policies for the detection of polymorphic blending attacks. Hidden Markov models, artificial intelligence techniques, neural networks, data mining methodologies and many other techniques have been proposed for the detection and mitigation of these advanced ransomwares. It is the need of the hour that anti-malware organisations update their policies and algorithms to detect these malwares so as to save the organisations from the lurking fear of these highly dangerous malwares and ransomwares.

References

1. Ahmadian, M.M., Shahriari, H.R., Ghaffarian, S.M.: Connection-monitor & connection-breaker: a novel approach for prevention and detection of high survivable ransomwares. In: Proceedings of 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC). IEEE, Iran, Rasht, 8–10 Sept
2. Gazet, A.: Comparative analysis of various ransomware virii. J. Comput. Virol. Springer, France **6**, 77–90 (2010)
3. Weckstén, M., Frick, J.: A novel method for recovery from crypto ransomware infections. In: Proceedings of the 2nd IEEE International Conference on Computer and Communications (2016)
4. Tseng, A., Chen, Y., Kao, Y.: Deep learning for ransomware detection. IEICE Tech. Rep. **116**, 87–92 (2016)
5. Lozada, L.: Ransomware: analyzing the impact on healthcare and the economy. Doctoral dissertation, Utica College (2017)
6. Ganorkar, S.S., Kandasamy, K.: Understanding and defending crypto-ransomware. ARPN J. Eng. Appl. Sci. **12**, 3920–3925 (2017)
7. Mercaldo, F., Nardone, V., Santone, A.: Ransomware inside out. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 628–637. IEEE, Aug 2016
8. Cabaj, K., Gawkowski, P., Grochowski, K., Osojca, D.: Network activity analysis of CryptoWall ransomware. Przegl. Elektrotechniczny **91**(11), 201–204 (2015)
9. Ferrante, A., Malek, M., Martinelli, F., Mercaldo, F., Milosevic, J.: Extinguishing Ransomware-a Hybrid Approach to Android Ransomware Detection
10. Certin, M.: Wannacry/WannaCrypt Ransomware. In: Cyber Swachhta Kendra. http://www.cyberswachhtakendra.gov.in/alerts/wannacry_ransomware.html. Accessed 24 July 2017
11. Cert, E.: WannaCry Ransomware Campaign Exploiting SMB Vulnerability. Retrieved from Cert Europa Website: https://cert.europa.eu/static/SecurityAdvisories/2017/CERT-EU-SA2017-012.pdf (2017)
12. McAfee Labs. Further Analysis of WannaCry Ransomware. In: securing tomorrow, today. https://securingtomorrow.mcafee.com/mcafee-labs/analysis-wannacry-ransomware/ (2017). Accessed 15 July 2017
13. Ma, W., Duan, P., Liu, S., Gu, G., Liu, J.C.: Shadow attacks: automatically evading system-call-behavior based malware detection. J. Comput. Virol. **8**(1), 1–13 (2012)
14. https://msdn.microsoft.com/en-us/library/bb742610.aspx
15. Akritidis, P., Markatos, E.P., Polychronakis, M., Anagnostakis, K.: Stride: polymorphic sled detection through instruction sequence analysis. In: IFIP International Information Security Conference, pp. 375–391. Springer, Boston, MA, May 2005

16. You, I., Yim, K.: Malware obfuscation techniques: a brief survey. In: 2010 International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA), pp. 297–300. IEEE, Nov 2010
17. Luo, X., Warkentin, M.: Developments and defenses of malicious code. Encycl. Multimed. Technol. Netw. **3** (2008)
18. Luo, X., Liao, Q.: Ransomware: a new cyber hijacking threat to enterprises. In: Handbook of Research on Information Security and Assurance, pp. 1–6. IGI Global (2009)

Improved EMD Local Energy with SVM for Fault Diagnosis in Air Compressor

Seetaram Maurya, Vikas Singh, Narendra Kumar Dhar and Nishchal K. Verma

Abstract The data generated from the machines are generally nonlinear and non-stationary in nature. Extracting relevant information from the data plays a major role in the fault diagnosis. This paper proposes local energy based feature extraction technique derived from improved empirical mode decomposition. Relevancy of feature is examined by correlation method. Support vector machine is used for classification of features. The proposed approach is compared with full signal energy using Hilbert transform on EMD. Improved empirical mode decomposition is used for decomposing acoustic signal into intrinsic mode function in lesser time compared to conventional EMD. Acoustic signals are acquired from most sensitive position of air compressor. Although acoustic signal-based machine health monitoring has not been applied to same extent as vibration signal. In this paper, acoustic signal has been used because of its advantage. The experimental results show the acceptable levels of average accuracy.

Keywords Fault diagnosis · Improved empirical mode decomposition (IEMD) Local energy · Support vector machine (SVM) · Relevant features extraction Acoustic signal

S. Maurya (✉) · V. Singh · N. K. Dhar · N. K. Verma
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: seetaram@iitk.ac.in

V. Singh
e-mail: vikkyk@iitk.ac.in

N. K. Dhar
e-mail: nardhar@iitk.ac.in

N. K. Verma
e-mail: nishchal@iitk.ac.in; nishchal.iitk@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_7

1 Introduction

Generally, rotating machines operate in stringent working conditions and are therefore subject to unexpected failure. The unexpected failures may cause a ruinous accident and financial losses [1, 2]. Fault diagnosis prevents unexpected failure, results in reduction of maintenance cost, and increases machinery lifetime. Although installation of fault monitoring system results in extra cost due to sensors and circuits. Extra computational effort is also needed in signal processing and decision making but the increase in cost is compensated by improving the reliability and security.

In [3, 4], authors give widespread ideas about the use of acoustic signals for various kind of fault diagnosis. These paper also mention some merits of acoustic signal compared to vibration signals. The merits are less sensitivity to resonances due to structure and mechanical noises. Acoustic signals are sensitive to disruptions due to faults.

Most of the time, feature extraction is done to obtain relevant information from the signals. Features are the parameters derived from the signals that indicate faults present in machines. Basically, feature extraction is done in three domain: time, frequency, and time-frequency. In [5, 6], researchers present time domain methods of feature extraction and have successfully applied to roller bearing. Some frequency domain methods are used for diagnosis of bearing faults in [7, 8]. In [9], researchers propose condition monitoring based on pattern analysis framework. In [10], authors present health monitoring of air compressor using open platform communication technology. As the signals generated by rotating machines are nonlinear and non-stationary and energy generated by machinery faults is low, time-frequency approach is effective. A comparative study of time domain, frequency domain, and time-frequency domain methods are discussed in [11–13]. Empirical mode decomposition is having self-adaptive characteristic of filtering, and hence it is a versatile tool for acoustic signal analysis in machinery fault diagnosis.

In the literature, researches have also applied statistical methods for fault detection, diagnosis and prognosis are artificial neural network (ANN), principal component analysis (PCA), and more recently SVM. SVM is based on the principle of structural risk minimization based on statistical learning by Vapnik and Chervonenkis. Boser et al. in [14] introduce SVM as machine learning algorithm for classification of data. In this paper, IEMD is used for decomposing acoustic signal into IMF in reduced time compared to conventional EMD and IEMD local energy is used for extraction of feature vector. SVM is used as an intelligent tool for classification of faults in air compressor. In [15–19], sensitive position for acquisition of acoustic signals from air compressor has been discussed in detail. A generic flow diagram of proposed methodology for fault diagnosis of air compressor is shown in Fig. 1. In Fig. 1, healthy represents data acquisition when air compressor is running in normal condition whereas outlet valve (LOV), leakage outlet valve (LIV), piston in faulty condition.

The remainder of the paper is organized as: Sect. 2 describes data acquisition from sensitive positions and steps of pre-treatment. Section 3 presents IEMD method for

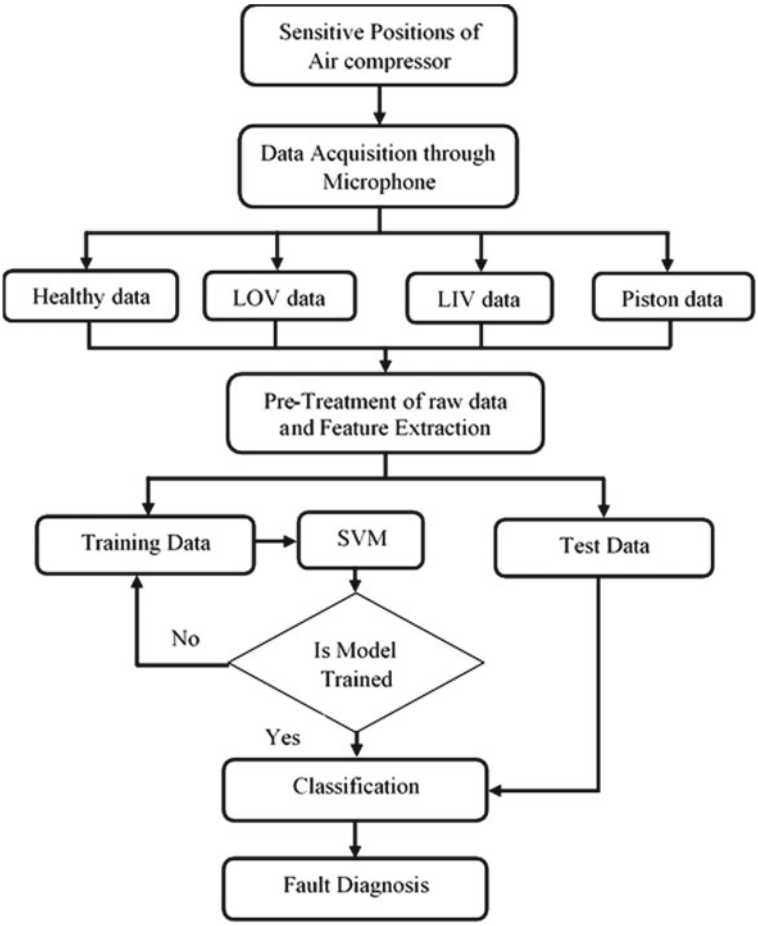


Fig. 1 Flowchart of the proposed methodology: LOV and LIV stand for leakage outlet valve and leakage inlet valve respectively

feature extraction. SVM method and fault diagnosis procedures are discussed in Sect. 4. Section 5 provides insight into experimental results for fault diagnosis of air compressor. Finally, conclusion of the paper is drawn in Sect. 6.

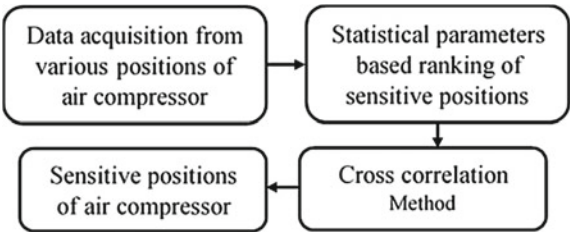


Fig. 2 Flowchart for finding sensitive positions of air compressor

2 Data Acquisition and Pre-treatment

2.1 Data Acquisition

Recording of data is the first and foremost step for fault diagnosis of air compressor as in any kind of machinery fault diagnosis. The acoustic signals have been recorded using unidirectional microphone. Raw acoustic signals generated from the air compressor under healthy condition and in different faulty conditions are recorded. Data acquisition has been done from most sensitive positions of air compressor. Figure 2 shows flowchart for finding sensitive position based on statistical parameter analysis and correlation method [15].

2.2 Pre-treatment

Most of the time, raw signals recorded from air compressor may suffer from noise and some other unwanted signals. To reduce these kind of unwanted effects, pre-treatment is needed. In this paper, filtering, clipping, smoothing, and normalization are performed on raw data.

3 IEMD Method and Feature Extraction

In this section, EMD algorithm [20] is described and calculation of local mean using envelope method of extrema is replaced by average method [21] for the successive extrema known as IEMD. The basic idea of EMD algorithm is to decompose the raw signal into IMFs. The signal can be decomposed into its IMFs if it satisfies the following two conditions [20].

1. Number of zero crossing and extrema must be either equal or differ by one in complete data set.
2. In complete data set at any point, the mean value of envelope defined by local minima and local maxima is zero.

To extract IMFs from a given signal, Huang suggested the decomposition process as given in Algorithm 1.

Algorithm 1 Decomposition Process

1. All local maxima and local minima are identified, then connect it using a cubic apline line as upper envelope $s_u(t)$ and lover envelope $s_l(t)$.
2. Find local mean $\mu_1(t)$ as given below:

$$\mu_1(t) = \frac{s_u(t)+s_l(t)}{2} \tag{1}$$

3. The first IMF of raw signal is computed as follows:

$$d_1(t) = c(t) - \mu_l(t) \tag{2}$$

   Here, $c(t)$ is the raw signal.
4. If $d_1(t)$ is not IMF, consider $d_1(t)$ as a raw signal and repeat this process until $d_1(t)$ is an IMF. The stopping criterion is the difference between the number of extrema and zero crossings becomes 0 or 1.
5. If the first component of IMF is obtain, find the residue signal $r_1(t)$ by subtracting first component from raw signal.

$$r_1(t) = c(t) - d_l(t) \tag{3}$$

6. To find second IMF component $d_2(t)$ as given below, consider $r_1(t)$ as raw signal and repeat the above process.

$$d_2(t) = r_1(t) - r_2(t) \tag{4}$$

7. Repeat above process number of times until residual signal becomes a monotonic function. Then n-IMF can be written as follows:

$$\begin{aligned} d_1(t) &= c_1(t) - r_1(t) \\ d_2(t) &= r_1(t) - r_2(t) \\ d_3(t) &= r_2(t) - r_3(t) \\ &\vdots \\ d_n(t) &= r_{(n-1)}(t) - r_n(t) \end{aligned} \tag{5}$$

   Adding both side of equation (5) accordingly, we can find

$$c(t) = \sum_{j=1}^{n} d_j(t) + r_n(t) \tag{6}$$

   This way, decomposition process is completed and n-IMF components are $d_1(t)$, $d_2(t)$, $d_3(t)$.....$d_n(t)$.

3.1 Average Method of Decomposition

The detailed mathematical explanation of this method is given in [21]. Defining local mean operator is defined as below:

$$\mu(q(t)) = mean(q(t)) \tag{7}$$

$mean(q(t))$ is the operation of computing local mean. Some properties of local mean operation is given below:

(a) $\mu(c(t) \pm q(t)) = \mu(c(t)) \pm \mu(q(t))$
(b) $\mu(c(t)q(t)) = \mu(c(t))\mu(q(t))$
(c) $\mu^n(c(t)) = \mu \ldots \mu(c(t))$

The basic properties of local mean operation will be used in further of expressions. Using Eq. (5), we can write

$$\begin{aligned} r_n(t) &= r_{(n-1)}(t) - d_n(t) \\ &= c(t) - \sum_{j=1}^{n} h_j(t) \end{aligned} \tag{8}$$

Using the basic properties of local mean operators and Eq. (7) can be written as below:

$$d_1(t) = \mu^{(n-1)}(c(t)) - \mu^n(c(t)) \tag{9}$$

The key idea behind EMD method is computing the local mean. Huang has given an effective envelope technique for computation of local mean, but spline interpolation must be done two times to find upper and lower envelopes. Therefore, this process of decomposing signal is a time consuming one. In improved technique, only extrema is calculated and local mean is arrived at by joining all average means between two consecutive extrema by cubic spline. The method is given in Algorithm 2.

Algorithm 2 IEMD Process

1. All local extrema is identified.
2. Find all average means between two successive extrema as given below:

$$\mu_k = \frac{1}{n_{(k+1)} - n_k} \sum_{n=n_k}^{n_{(k+1)}} c(n\Delta t) \tag{10}$$

where $k = 1, 2, 3, \ldots N-1$. N is the number of extrema, Δt represents sampling interval, $n_{k+1}\Delta t$ and $n_k \Delta t$are the period of two successive extremas. The time instant with respect to average mean can be estimated as follows:

$$t'_k = \frac{\mu_k - c(n_k \Delta t)}{c(n_{(k+1)}\Delta t) - c(n_k \Delta t)} n_k \Delta t + \frac{\mu_k - c(n_{(k+1)}\Delta t)}{c(n_k \Delta t) - c(n_{(k+1)}\Delta t)} n_{(k+1)} \Delta t \tag{11}$$

3. By connecting all average means $\mu_k(t'_k)$ using cubic spline, consider as local mean $\mu(t)$ of raw data. This data is to be decomposed.
4. After computation of local mean, decomposition steps are same as given by Huang.

$\mu_i(t)$ is computed every time, whereas cubic spline method is applied only once. While using envelope method it is applied twice. Therefore, IEMD method takes less computational time.

4 SVM and Fault Diagnosis Procedure

4.1 SVM Method

SVM is a very effective technique for classification and regression of any kind of data [22]. SVM first started as a binary classifier that separates two classes with maximum achievable margin [23]. For nonlinearly separable problem, a kernel trick is useful. There are different versions of SVM used in several applications like text classification, image classification, time-series prediction [24–26]. There are some popular kernel like linear, polynomial, sigmoid, and radial basis function (RBF) [27]. In this paper, linear and radial basis function kernels are used. A general equation of hyperplane has been given in Eq. (11), where w is a weight matrix and b is a constant bias. The expression for function kernel used in this paper is given as follows:

linear kernel : $K_{linear}\langle x_i, y\rangle = x_i.y$

radial basis function kernel : $K_{RBF}\langle x_i, y\rangle = e^{[-\gamma\|x_i - y\|^2]}$

$$f(x) = \langle w, x\rangle + b \tag{12}$$

The final expression of hyperplane is given by Eq. (12). Based on kernels, the decision function of the classifier is estimated [28].

$$f(x) = sgn(\sum_{i=1}^{n} y_i \alpha^* K\langle x_i, x\rangle + b) \tag{13}$$

4.1.1 Cross-Validation

With RBF kernel function, there is a need of appropriate parameter to make sure that classifier is accurately predicting the test data. Finding an optimal parameter could be done using cross-validation [28]. Cross-validation algorithm is used to validate the model and to find the predictive model accuracy. In this paper, fivefold cross-validation is used to check misclassification rate in all test data.

4.2 Fault Diagnosis Procedure

The detailed steps for fault diagnosis are as follows:

1. Raw acoustic signals are collected from air compressor under healthy condition and three faulty condition.
2. Raw acoustic signals are pre-treated. Pre-treated acoustic signals under healthy and faulty condition are shown in Fig. 3.

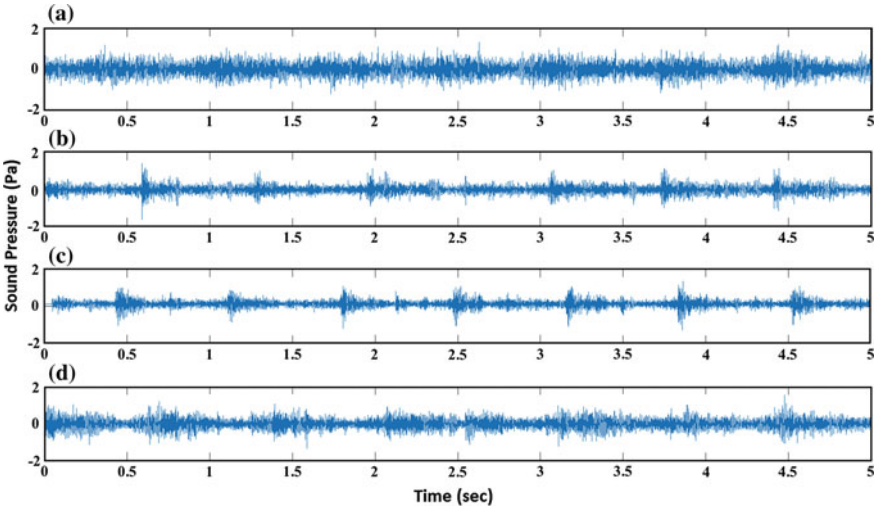


Fig. 3 Pre-treated acoustic signals. The x-axis is the time of recording in seconds, and y-axis is the sound data in Pascals. Healthy acoustic signals are shown in (**a**). Three types of acoustic signal faults: LOV, LIV, and piston faults are shown in (**b**), (**c**), (**d**), respectively

3. The pre-treated acoustic signals are decomposed into n-IMF and. For feature extraction, most dominant IMFs are considered.
4. Find all local energy W_i of most dominant IMF.

$$\mathtt{W}_i = \int_{-\infty}^{+\infty} |d_i(t)|^2 dt \quad \forall \ i = 1 \cdots k \tag{14}$$

5. Construct a feature vector F containing W_i as element.

$$F = [W_1, W_2, W_3, \ldots W_k] \tag{15}$$

Sometimes energy is very large, so the feature vector is normalized as below:

$$F' = \frac{F}{\sum_{i=1}^{k} W_i} \tag{16}$$

where F' is the normalized feature vector.

6. Construct a classifier model. As discuss in Sect. 4.1, linear and RBF kernel are used. Cross-validation has been performed to obtain optimal parameter.

5 Experimental Results

An experiment on air compressor has been done. The technical description of air compressor used for experimentation are given below:

- Air-pressure range:
  0–2224.108 Pa, 0–35 kg/cm^2
- Induction motor:
  5 h_P, 5 A, 415 V, 50 Hz, 1440 rpm
- Pressure switch:
  - Range: 100–213 lb/in.^{2}, 7–15 kg/cm^2
  - Type: *PR*-15

For experimental analysis, microphone (Ahuja UTP-30) with sensitivity of 7.0 mv/Pa, frequency range 10–15 kHz, and supply voltage 1.5 V DC is used. The computations have been performed on a desktop computer: *i*5 2320 processor and RAM are 8 GB. Acoustic signals have been recorded from air compressor in four different states. The details of four different states of air compressor are:

- Healthy is a state of air compressor when it is running in normal condition means free from any kinds of faults.
- LOV is the state when leakage takes place in outlet valve of the air compressor.
- LIV is a state of air compressor when inlet valve is damaged; therefore, leakage takes place.
- Piston is a faulty state of air compressor because of loosening the piston ring; therefore leakage takes place.

Acoustic signal has been used for the experiment because of some merits compared to vibration signals. All collected acoustic data from four states of the air compressor were pre-treated with pre-treatment module discussed in Sect. 3. Further the pre-treated data are used for feature extraction. IEMD method used for feature extraction takes lesser computational time as compared to conventional EMD methods. Tables 1 and 2 show the comparison of average accuracy percentage of proposed method of feature construction and full signal energy feature (FSEF) method [29] of feature construction using Hilbert transform on acoustic data set [30]. The percentage average accuracy of classification has been calculated with fivefold cross-validation. As given in Table 2, top five features are selected based on ranking. Ranking of features was done by individual training error ranking (ITER). In ITER ranking of features, two features are compared and error is calculated. Feature which is having least error ranks first. Table 2 shows better average accuracy as compared to random features selected in Table 1.

Table 1 Comparison of average accuracy percentage of proposed method and FSEF methodology without features selection

Air compressor states	Features	FSEF method Accuracy		Proposed method accuracy	
		Linear Kernel	RBF Kernel	Linear Kernel	RBF Kernel
Healthy-Healthy	10	100.00	100.00	100.00	100.00
Healthy-LIV	10	96.44	96.44	**97.33**	**97.33**
Healthy-LOV	10	76.44	76.45	**87.33**	**87.33**
Healthy-Piston	10	85.56	85.33	**88.89**	**88.67**

Table 2 Comparison of average accuracy percentage of proposed methodology and FSEF method with selected features

Air compressor states	Features	FSEF method accuracy		Proposed method accuracy	
		Linear Kernel	RBF Kernel	Linear Kernel	RBF Kernel
Healthy-Healthy	5	100.00	100.00	100.00	100.00
Healthy-LIV	5	96.67	96.44	**97.56**	**97.33**
Healthy-LOV	5	77.78	77.12	**88.67**	**88.00**
Healthy-Piston	5	88.56	87.44	**88.89**	**90.67**

6 Conclusions

This paper presents real-time fault diagnosis in air compressor based on IEMD local energy method and SVM using acoustic signal. Development of fault diagnosis model includes data acquisition from sensitive positions in four different states of air compressor, pre-treatment, feature extraction, and classification. For feature extraction, IEMD method was used which takes lesser computational time compared to conventional EMD. SVM method is used for classification. In this paper, two types of kernel namely linear and RBF were used. Proposed model has been tested with random features and with selected features based on ITER ranking. In both the cases, the proposed model has satisfactory performance.

References

1. Verma, N.K., Sreevidya, K.: Cost benefit analysis for maintenance of rotating machines. In: IEEE International Conference on Prognostics and Health Management, pp. 1–7. Austin, USA, 22–25 Jun 2015
2. Verma, N.K., Ghosh, A., Dixit, S., Salour, A.: Cost-benefit and reliability analysis of prognostic health management systems using fuzzy rules. In: IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (IEEE WCI 2015), pp. 1–9. India, 14–17 Dec 2015
3. Li, R., He, D., Bechhoefe, E.: Investigation on fault detection for split torque gearbox using acoustic emission and vibration signals. In: Annual Conference of the Prognostics and Health Management Society, pp. 1–11 (2009)
4. Holroyd, T.J.: Acoustic Emission and Ultrasonics Monitoring Handbook. Coxmoor Publishing Company (2000)
5. Martin, H.R.: Statistical moment analysis as a means of surface damage detection. In: Proceedings of the International Modal Analysis Conference, pp. 1016–1021 (1989)
6. Peng, Y.H., Richard, Y.: Wavelet analysis and envelope detection for rolling element bearing fault diagnosis their effectiveness and flexibilities. J. Vib. Acoust. Trans. ASME 303–310 (2001)
7. Volker, E., Matin, H.R.: Application of kurtosis to damage mapping. In: Proceedings of the International Modal Analysis Conference, pp. 629–633 (1986)
8. Zheng, G.T., Wang, W.J.: A new cesptral analysis procedure of recovering excitations for transient components of vibration signals and applications to rotating machinery condition monitoring. J. Vib. Acoust. **123**(2), 222–229 (2001)
9. Verma, N.K., Kumar, P., Sevakula, R.K., Thirukovalluru, R.: Pattern analysis framework with graphical indices for condition based monitoring. IEEE Trans. Reliab. (2017) (Accepted)
10. Verma, N.K., Sharma, T., Maurya, S., Singh, D., Salour, A.: Real-time monitoring of machines using open platform communication. In: IEEE International Conference on Prognostics and Health Management (2017)
11. Benbouzid, M.E.H.: A review of induction motors signature analysis as a medium for faults detection. IEEE Trans. Ind. Electron. **47**(5), 984–993 (2000)
12. Henriquez, P., Alonso, J.B., Ferrer, M.A., Travieso, C.M.: Review of automatic fault diagnosis systems using audio and vibration signals. IEEE Trans. Syst. Man. Cybern. **44**(5), 642–652 (2014)
13. Verma, N.K., Sevakula, R.K., Dixit, S., Salour, A.: Intelligent condition based monitoring using acoustic signals for air compressors. IEEE Trans. Reliab. **65**(1), 291–309 (2016)
14. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Workshop on Computational Learning Theory, pp. 144–152 (1992)
15. Verma, N.K., Kumar, P., Sevakula, R.K., Dixit, S., Salour, A.: Ranking of sensitive positions based on statistical parameters and cross correlation analysis. In: Sixth International Conference on Sensing Technology (ICST), pp. 815–821 (2012)
16. Verma, N.K., Kumar, P., Sevakula, R.K., Dixit, S., Salour, A.: Ranking of sensitive positions based on statistical parameters and cross correlation analysis. In: 6th IEEE International Conference on Sensing Technology, Kolkata, India, pp. 815–821 (2012)
17. Verma, N.K., Singh, N.K., Sevakula, R.K.: Ranking of sensitive positions using empirical mode decomposition and Hilbert transform. In: IEEE Conference on Industrial Electronics and Applications, pp. 1926–1931 (2014)
18. Verma, N.K., Sevakula, R.K., Dixit, S., Salour, A.: Ranking of sensitive positions using statistical and correlational analysis. Int. J. Smart Sens. Intell. Syst. **6**(4), 1745–1762 (2013)
19. Verma, N.K., Jagannatham, K., Bahirat, A., Shukla, T.: Finding sensitive sensor positions under faulty condition of reciprocating air compressors. In: International Conference on IEEE Recent Advances in Intelligent Computational Systems, Trivandrum, pp. 242–246 (2011)
20. Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and

non-stationary time series analysis. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. **1971**(454), 903–995 (1998)
21. Du, Q., Yang, S.: Improvement of the EMD method and applications in defect diagnosis of ball bearings. Meas. Sci. Technol. **17**(8), 2355–2361 (2006)
22. Scholkopf, B., Smola, A.J.: Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
23. Vapnik, V.N.: An overview of statistical learning theory. IEEE Trans. Neural Netw. **10**(5), 988–999 (1999)
24. Chapelle, O., Haffner, P., Vapnik, V.N.: Support vector machines for histogram-based image classification. IEEE Trans. Neural Netw. **10**(5), 10551064 (1999)
25. Stanislaw, O., Hoai, L.T., Markiewicz, T.: Support vector machine based expert system for reliable heartbeat recognition. IEEE Trans. Biomed. Eng. **51**(4), 582–589 (2004)
26. Sapankevych, N.I., Sankar, R.: Time series prediction using support vector machines: a survey. IEEE Comput. Intell. Mag. **4**(2) (2009)
27. Vapnik, V., Cortes, C.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
28. Yin, S., Gao, X., Karimi, H.R., Zhu, X.: Study on support vector machine-based fault detection in Tennessee Eastman process. In: Abstract and Applied Analysis. Hindawi Publishing Corporation (2014)
29. Huang, J., Hu, X., Geng, X.: An intelligent fault diagnosis method of high voltage circuit breaker based on improved EMD energy entropy and multi-class support vector machine. Electr. Power Syst. Res. **81**(2), 400–407 (2011)
30. Intelligent Data Engineering and Automation (IDEA) Laboratory, Indian Institute of Technology, Kanpur [online]. http://www.iitk.ac.in/idea/datasets/download-ACdata.php

Simultaneous Structure and Parameter Learning of Convolutional Neural Network

Soniya, Sandeep Paul and Lotika Singh

Abstract This paper provides a solution to select a suitable architecture of convolutional neural network (CNN). A hybrid evolutionary gradient descent (HyEGD) approach is proposed to automatically evolve the architecture of CNN. The evolution of the structure is done using compact genetic algorithm (cGA) by optimizing the number of filters in each layer, and simultaneously, the associated weight parameters are tuned by stochastic gradient descent (SGD). This brings forth an effective way to search the solution space seamlessly integrating both *exploration*, spearheaded by cGA, and the *exploitation*, naturally done by SGD. Moreover, using HyEGD approach, the user specified architecture can also be evolved trading-off between two objectives—*network performance* on one side and *network size* on the other side. Experiments to illustrate the salient features of the HyEGD approach are performed on two benchmark problems: COIL-20 dataset and MNIST dataset. The results clearly highlight the powerful capability of generating architectures based on the required performance and size of network.

Keywords Convolutional neural network · Structure and parameter learning
Hybrid evolutionary gradient descent

1 Introduction

Convolutional neural network (CNN) has emerged as one of the prominent deep neural networks providing high performance on a wide range of application domains [9]. The architecture of CNN comprises of many interconnected functional layers

Soniya · S. Paul (✉) · L. Singh
Dayalbagh Educational Institute, Agra, Dayalbagh, India
e-mail: spaul@dei.ac.in

Soniya
e-mail: dei.soniya@gmail.com

L. Singh
e-mail: lotikasingh@dei.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_8

like convolutional layer, pooling layer, activation layer and fully connected layer. Specifically, each convolutional layer consists of a number of filters, which in turn are the block of weights. These weights are tunable parameters and are learned typically by some variant of gradient descent law. The performance of the CNN largely depends on, firstly the *structure*—number of convolutional layers and its associated filters, and secondly the *parameters*—weights of filters. The design of architecture more-or-less still is based on heuristic rules. For best performance, this heuristically designed architecture also varies with applications making the selection of a suitable architecture even more tough. Thus, efforts are made towards automatic designing of the architecture of CNN.

Evolutionary algorithms (EAs) have shown significant success for optimization problems [1, 3]. Some of the well-known evolutionary algorithms are genetic algorithm [5], particle swarm optimization [17], differential evolution [4], ant colony optimization [8] and others. All of these algorithms are population-based methods and are computationally expensive in terms of memory requirements and fitness function evaluation of the population. Another domain of EAs is compact evolutionary algorithms (cEAs) which are based on the notion of probability vector [6]. These compact algorithms work with virtual population as compared to conventional EAs and hence curtail the memory requirements. The compact genetic algorithm [6], real-coded genetic algorithm [13], compact differential evolution [14] are well-known compact algorithms.

The integration of deep networks with evolutionary algorithms is in its nascent state. Some advantages of evolutionary algorithms are as follows: they are robust and capable of exploring both structure and parameter simultaneously [20]. There are few papers in the literature which have contributed in this direction. Rere et al. [19] used three metaheuristic approaches as simulated annealing, differential evolution and harmony search for learning of CNN. They have used MNIST and CIFAR dataset for classification and reported accuracy of 98.65% for MNIST dataset using simulated annealing approach.

Recently, Xie in [23] has proposed a genetic CNN model. This model uses genetic algorithm for evolving CNN. The number of filters in each convolutional layer of network was kept fixed. They divided the network into S stages, and two consecutive stages were connected by a pooling layer. The number of convolutional layers in the stage S_i had been evolved using fixed-length binary string method. During each generation, different architectures were evolved using genetic algorithm and training of each architecture was done via gradient descent.

Real et al. [18] proposed a new method to discover the best architecture using genetic algorithm. For genetic algorithm, they used the population of size 1000 individuals. Each individual represents an architecture which is encoded as a graph where vertices are represented by rank-3 tensor or activations and edges represent identity connection or convolutions. The algorithm creates multiple complex networks from scratch without putting any constraint on a number of layers.

This paper proposes a solution for evolving the architecture of CNN automatically using a hybrid evolutionary gradient descent (HyEGS)approach. Compact genetic algorithm has been used to explore the solution space which consists of the all

possible network structures, while the weights of filters of CNN are learned by stochastic gradient descent law. Inspired from [16], the structure of CNN is evolved by finding the suitable number of filters in each convolutional layer. The choice of compact evolutionary algorithm makes HyEGS approach fast and scalable. To the best of our knowledge, this is the first paper which integrates CNN with compact genetic algorithm (cGA) for structure learning of the network. One of the major strength of HyEGS approach is to evolve varying architectures based on the need of *network performance* or *network size*.

The rest of the paper is organized as follows: Sect. 2 gives a detailed description of the proposed HyEGD approach; experiments and results are discussed in Sect. 3; Sect. 4 presents conclusion and future scope.

2 HyEGD: Hybrid Evolutionary Gradient Descent Approach

The proposed approach, HyEGD, presents a novel method for automatically discovering the number of filters of CNN. The integration of persistent compact genetic algorithm [2] with CNN has been used in this approach. The persistent compact genetic algorithm preserves a copy of the best solution as opposed to compact genetic algorithm where the best solution is searched at each iteration. The filters of network and hence the network structure is evolved by persistent compact genetic algorithm, and learning of the weight parameters is done using stochastic gradient descent algorithm. In the proposed approach, the string S consists of two parts: binary and real-valued parts.

The binary part represents the presence or absence of filters in a network with the binary numbers 1 and 0, respectively. The real part of the string represents weights of a network. The binary part includes a total number of filters, while the weights values of each filter are encoded in real part of the string. Figure 1 shows a pictorial representation of structure of both binary and real-valued parts of the string S.

Consider a CNN as shown in Fig. 1 which has total four convolutional layers. The first two convolutional layers are followed by pooling and ReLU layer. The third and fourth convolutional layers are followed by ReLU only. The binary part encodes the presence or absence of filters for each convolutional layer as depicted in Fig. 1. The length of binary part (l) can therefore be formulated as follows:

$$l = \sum_{i=1}^{L} f_i, \tag{1}$$

where L is the total number of convolutional layers; filter_dimension represents the dimension of filter; and f_i is number of filters in the ith convolutional layer. Similarly, length of the real part (m) can be written as

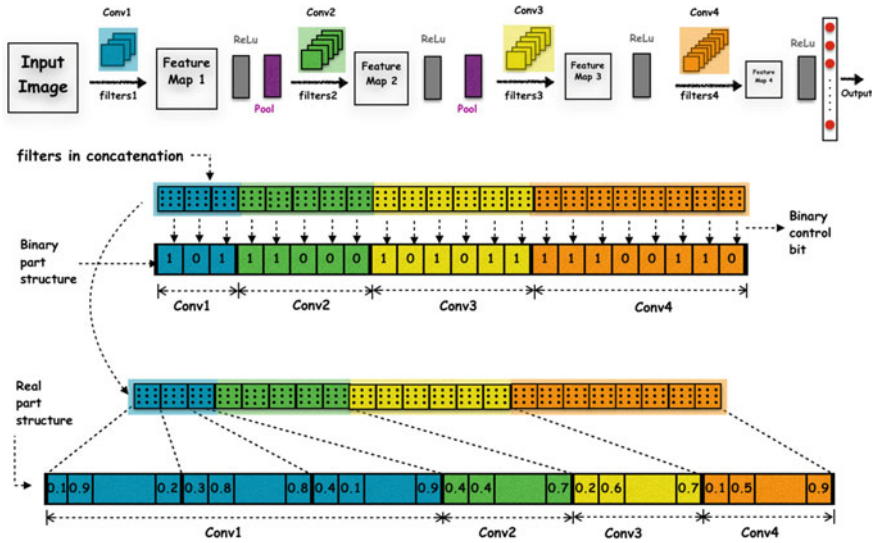


Fig. 1 Representation of a typical convolutional neural network by a string comprising concatenated binary and real part. Each bit of binary part of string reflects the presence or absence of a filter of convolutional layer. Each value of real part of string represents the weights of each filter

$$m = \sum_{i=1}^{L} f_i * (filter_dimension) \tag{2}$$

The algorithm initializes the following: binary probability vector, pop_size, learning rate (lr), number of iterations (itt), and size of batch ($batch_size$). The binary probability vector (PB) is initialized with value 0.5 which signifies that each filter of network $\mathcal{N}$ has 50% chance that it will be a part of the network architecture. The pop_size defines the step size used for updating PB. The learning rate (lr), number of iterations (itt), and size of batch ($batch_size$) are the parameters of stochastic gradient descent (SGD) method which are used during training of network architecture. The F_{min} and F_{max} values give a range of upper and lower limits of the number of filters being used. The optimal number, however, is being evolved using the hybrid cGA and SGD where cGA is used for exploration of the search space and SGD is used for exploitation. The pseudocode of the proposed approach is given in Algorithm 1.

The flexibility of the algorithm to choose architectural economy over performance is an important aspect of the proposed approach. A customized cost function plays an important role in guiding the learning process which is defined as follows [16, 21]:

$$Costfunction = \alpha * (\#filters) + (1 - \alpha) * (error) \tag{3}$$

where $\#filters$ is the total number of filters of CNN and $error$ is the misclassification error.

The parameter α is architectural economy parameter and can be consider as user-defined parameter which provides the trade-off between performance and weight parameters of the network. For larger value of α (e.g. 0.9), more weightage is given to the number of filters present in the network. The search therefore moves towards finding network structure with less number of filters (hence less weights in the network) and hence a smaller network architecture in terms of the number of weights used. This, however, is at the cost of network performance or classification accuracy. For smaller values of α, the (1-α) value increases which gives more weightage to the second term in the cost function which is classification error of the network. The search in this case therefore moves towards finding network structure with lower classification error (or higher network performance in terms of classification accuracy) without any consideration of reducing the number of weights or filters of the network. This in other words leads to a network with more number of filters and hence more weights and therefore a bigger network. Thus, the parameter α provides flexibility between network performance and architecture for a given problem.

3 Experiments

The proposed approach, HyEGD, has been tested on two benchmark datasets: COIL-20 and MNIST. The Columbia Object Image Library (COIL) dataset [15] has two variants, namely COIL-20 and COIL-100. The images in the dataset show complex geometric and reflectance characteristics. COIL-20 has 20 object classes. Each class has 72 grayscale images of size 128×128 of the object rotated at every 5 degree. It has total 1440 images. COIL-100 consists of total 7200 images and has 100 target classes. In the present study, only COIL-20 dataset has been used.

The MNIST dataset [11] has grayscale images of size 28×28, and it consists of 70,000 images of handwritten digits. As mentioned in [11], the dataset is divided into 60,000 training images and 10,000 test images. For both the datasets, our aim is of object recognition by using the proposed algorithm.

The algorithm was developed on PyTorch 0.1.12 platform. The simulations were performed on NVIDIA Tesla K80 GPU.

3.1 COIL-20 Dataset

The dataset is divided into non-overlapping training and test sets in the ratio 80%-20%. The training set has total 1160 images, and test set has 260 images. The architecture of CNN for COIL-20 dataset is as follows: Conv1-Pool1-Conv2-Pool2-Conv3-ReLU-Conv4-fc. The last layer (fc) is fully connected and has 20 neurons which represent target classes. Both pooling layers use max pooling with receptive

Algorithm 1 : Operational summary of the HyEGS approach

```
Given:
    A fixed network architecture (𝒩)
    A range of number of filters F_min,....,F_max for each convolutional layer of 𝒩

Initialize:
    binary probability vector (PB) with 0.5;
    pop_size; impacts the size of steps used for updating PB
      SGD parameters: lr = learning rate; impacts the size of steps used for updating weights of network
                      batch_size = size of batch; responsible for dividing dataset into number of batches
                      itt = number of iterations;

Begin:
    /* Generate initial string S_0 */
    B_0 = Generate(PB); binary part of string S_0 of length l
    Generate R_0; real part of string S_0 of length m
    S_0 = B_0 R_0; concatenated binary and real part

    𝒩(S'_0) = SGDTrain[𝒩(S_0)]; S'_0 = B_0 R'_0 [only R_0 is updated]
    C_0 = Cost[𝒩(S'_0)]; evaluate cost
    C_winner = C_0; S_winner = S'_0

    repeat {
            /* Generate string S_i */
            B_i = Generate(PB);
            Generate R_i;
            S_i = B_i R_i;

            𝒩(S'_i) = SGDTrain[𝒩(S_i)]; S'_i = B_i R'_i [only R_i is updated]
            C_i = Cost[N(S'_i)];

            /* Compare (C_i, C_winner) */
            if (C_i ≤ C_winner) then
                C_loser = C_winner; S_loser = S_winner
                Update: C_winner = C_i; S_winner = S'_i;
            else
                C_loser = C_i; S_loser = S'_i
            end

            /* Update binary probability vector (PB) */
            for k = 1:l   where l is length of PB as described in section 2
                if (S_winner(k) ≠ S_loser(k)) then
                    if (S_winner(k) == 1) then
                        PB(k) = PB(k) + 1/pop_size;
                    else
                        PB(k) = PB(k) - 1/pop_size;
                    end
                end
            end
        } while (PB == 0 || PB == 1)

End:
    The PB converges when all bit values become 0 or 1. The string
    S_winner represents the best architecture.
```

Table 1 COIL-20 Dataset: effect of α parameter

α	Total weights	Filters	Testing accuracy
0.1	1911	5-8-9-8-20	1.0000
0.3	1387	5-4-8-9-20	0.9964
0.5	827	3-2-7-7-20	1.0000
0.7	531	3-2-5-5-20	0.9607
0.9	307	3-3-2-3-20	0.7607

field of size 2×2 having stride of 2. Since the network is represented by fixed-length binary strings, there are upper and lower limits on the number of filters for each convolutional layer. For COIL-20 dataset, the upper and lower limits are set as 15-15-15-15-20 and 1-1-1-1-20, respectively. Filters of all four convolutional layer have been evolved by the proposed approach. Thus, the network has a total of 60 (=15 + 15 + 15 + 15) evolvable filters.

The proposed method starts with an architecture within the upper and lower limits. The network is encoded in binary string having the length of 60 bits with 1 and 0 indicating the presence and absence of a filter, respectively. The training of the network is done by gradient descent algorithm with learning rate as 0.0001 and batch size of 40 images for 100 iterations. The obtained architecture has 3-3-2-3-20 filters for $\alpha = 0.9$. For other values of α, different architectures were obtained and the results have been tabulated in Table 1. It can be observed from the table that for a large value of α (say 0.9), the obtained architecture has 3-3-2-3-20 filters and total 307 weight connections which gives the classification accuracy of 76.07%. However, for small value of α (say 0.1), the obtained architecture has 5-8-9-8-20 filters with 1911 weight connections which gives the classification accuracy of 100%. The experiments have been repeated five times for each value of α, and only best results are reported here.

It can therefore be inferred that with smaller value of α the classification rate is high because the weightage of second term in cost function is high. This is achieved at the cost of more weight connections. However, for larger value of α, the obtained architecture has less weight parameters but compromises with classification rate. Hence, the parameter α provides a flexibility in between architecture and performance of the network.

The distribution of filters in each convolutional layer is shown in column 3 of Table 1. With $\alpha = 0.1$, it can be seen that the number of weights used by the network for evaluation is 1911 which is very high as compared to 307 with $\alpha = 0.9$. Smaller α values focus on the performance, while greater values focus on architectural economy leading to lesser number of weights in the network. During search of architecture in the entire search space, different architectures have been evolved and are tabulated in Table 2. It can be observed from the table that as the number of epochs elapses

Table 2 COIL-20 Dataset: evolution of architectures for $\alpha = 0.5$

Epochs	Cost function	Filters	Weights
1	0.3057	6-10-5-10-20	1745
3	0.2864	5-8-7-9-20	1705
4	0.2481	5-5-4-8-20	940
8	0.2331	4-2-6-7-20	773
19	0.2322	2-3-7-8-20	965
23	0.2306	4-1-5-9-20	741
25	0.2281	3-2-6-9-20	895
26	0.2279	5-2-7-7-20	883
29	0.2259	4-2-6-9-20	923
30	0.2209	3-2-7-7-20	827
46	0.2120	3-2-7-7-20	827
53	0.2013	3-2-7-7-20	827

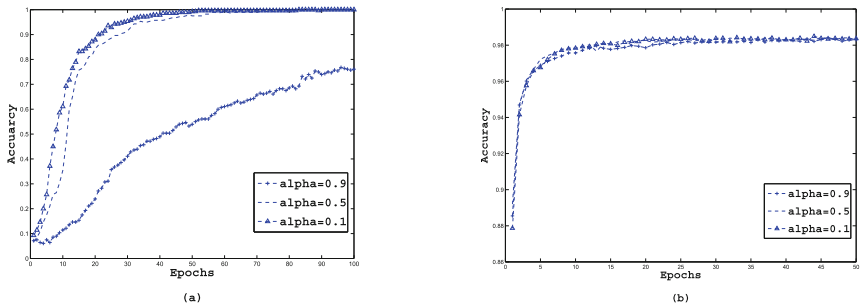


Fig. 2 Test accuracies of the obtained architecture for different values of architectural parameter, α: **a** COIL-20 dataset. **b** MNIST dataset

from 1 to 53 the value of cost function is decreasing, which indicates decrease in the classification error. At the same time, the number of filters has also evolved to smaller values from the initial filter values at epoch 1 (except at layer 3) in order to minimize total weights of the network. The interim changes in the architecture of the network also clearly illustrate the effectiveness of the proposed approach with respect to the evolution of the filter and hence the architecture of the network. Figure 2 shows the test accuracies of the obtained architectures for different values of α. As it is clear from the Fig. 2a that for $\alpha = 0.1$ and 0.5, with given set of parameters the obtained architectures have sufficient number of filters and thus the architectures have trained well in 100 epochs to classify all the test patterns correctly. However, with $\alpha = 0.9$, the network has less number of filters and hence gives the classification accuracy of just 77.07%. Thus, the architecture of the network is not sufficient.

Table 3 MNIST Dataset: effect of α parameter

α	Total weights	Filters	Testing accuracy
0.1	4842	4-8-10	0.9876
0.3	4251	4-7-10	0.9851
0.5	3220	5-5-10	0.9807
0.7	2226	2-4-10	0.9772
0.9	1144	2-2-10	0.9712

3.2 MNIST Dataset

The network used for MNIST dataset is as follows: Conv1-BatchNorm-ReLU-Pool1-Conv2-BatchNorm-ReLU-Pool2-fc. It has two convolutional layers followed by batch normalization, ReLU and max-pooling layer. The convolutional layers use receptive field of size 5×5 with stride 1, and both pooling layers use receptive field of size 2×2 with stride 1. The two convolutional layers have 20 filters in each. The last layer (fc) is fully connected and has ten target classes. Thus, filters of the architecture can be written as 20-20-10. For this dataset, the upper and lower limits on a number of filters for each convolutional layer are set as 20-20-10 and 1-1-10, respectively. The filters of first two convolutional layers are evolvable parameters in the algorithm. Thus, the network has a total of 40 (=20 + 20) evolvable filters.

The network starts evolving from an architecture within the upper and lower limits. In this case, the binary string has length of 40 bits. The training of each individual network is done via stochastic gradient descent algorithm with learning rate as 0.001 and batch size of 1000 images for 50 epochs. The obtained architecture for $\alpha = 0.1$ has 4-8-10 filters. For other values of α, the obtained architectures are tabulated in Table 3. The experiments have been repeated five times for each value of α, and only best results are reported here. It is clear from the table that for small value of α the accuracy of network is higher as compared to its large value. For $\alpha = 0.1$ and 0.9, the number of parameters is 4842 and 1144, respectively, as given in Table 3. The overall weight parameters are less for high value of α as compared to its small value. Thus, the user-defined parameter α helps in selection between two extremes—on one side the network performance and on the other side the economy of network architecture in terms of tunable parameters. Figure 2b shows the test accuracy of obtained architectures with different value of α for MNIST dataset.

With another network as Conv1-BatchNorm-ReLU-Pool1-Conv2-BatchNorm-ReLU-Pool2-Conv3-BatchNorm-ReLU-Pool3-fc, the HyEGD approach shows an accuracy of 99.47%. The architecture has three convolutional layer followed by batch normalization, ReLU and max-pooling layer. All three convolutional layers have receptive field of size 5×5 with stride 2, and all pooling layers use receptive field of size 2×2 with stride 1. The upper and lower limits on number of filters are set as 100-100-100-10 and 1-1-1-10, respectively. In this case, for $\alpha = 0.1$, the

obtained network has architecture of 38-36-39-10 with 73883 weight parameters and gives accuracy of 99.47%.

The HyEGD approach has been tested on both COIL-20 and MNIST datasets. It is observed that the approach is able to find architecture based on user's requirement of architectural economy or performance. As an example, the upper limit architecture for COIL-20 is taken as 15-15-15-15-20. It had total 6590 weight connections and showed 100% classification accuracy when trained using stochastic gradient descent. However, the obtained architecture by the HyEGD approach gives 827 weight parameters with 100% accuracy for $\alpha = 0.5$. In this case, the obtained network has around eight times less parameters as compared to the architecture with 15-15-15-15-20 filters. In case of $\alpha = 0.9$, the architecture has 307 parameters with 76.07% accuracy. Here, it is evident that for different of values of α, the HyEGD approach is giving architecture which has less number of weight connections as compared to initial architecture. Similarly for MNIST dataset, the initial architecture is taken as 20-20-10 with 20,350 weights parameters. The obtained architecture for $\alpha = 0.1$ has 4842 learning parameters and showing 4.2028 times reduction in weights parameters. However, for $\alpha = 0.9$, this number is 1144 which is much less as compared to 20,350 connections. Thus, it is clear that this approach is good for evolving filters without compromising much in classification accuracy.

A comparison between the best results of HyEGD approach and other existing methods based on their performances can be found in Table 4. For COIL-20 dataset, the HyEGD approach gives 100% classification accuracy. However, the accuracy of 72.40% was obtained by the approach which combined the k-means clustering with deep autoencoder [7]. In another work [12], the fully convolutional autoencoder-based clustering method achieved accuracy of 79.30%. Hence, it is clear that the proposed approach has achieved the good accuracy for the dataset. For MNIST dataset, the HyEGD approach has obtained an accuracy of 99.47% with 73883 learning parameters which is comparable with the results of [23]. Thus, it is clear that the HyEGD approach performed well for MNIST dataset.

Table 4 Comparison of HyEGD with other existing techniques for COIL-20 and MNIST dataset

Approaches	Accuracy (%)	Weights
COIL-20 dataset		
Huang et al. [7]	72.40	2,26,316
Li et al. [12]	79.30	2180
HyEGD approach	**100.00**	**827**
MNIST dataset		
Xie et al. [23]	99.66	–
Wang et al. [22]	99.20	27,858
LeCun et al. [10]	99.05	4,31,000
HyEGD approach	98.76	4842
HyEGD approach	99.47	73,883

4 Conclusion and Future Scope

This paper introduces a strategy to simultaneously evolve the architecture of deep neural network as well as learning the tunable parameters. The network architecture is evolved in terms of number of filters in each layer along with the training of weights by hybrid evolutionary gradient descent approach. The multi-objective fitness function is introduced and used to achieve seamless user-defined control over network architectural economy and network performance. Two important inferences supported by simulation studies are: firstly, for different values of user-defined architectural economy parameter α, the HyEGD approach is able to evolve economical architecture with less number of learning parameters without much compromising in classification accuracy; and secondly, the smaller value of α results in higher performance in contrast to the larger value of α which results in better network architectural economy. In either case, it has been found to yield superior performance with economical architecture as compared to other models in the literature. In future, other parameters like number of layers can also be considered for making the structure more adaptive which may lead to more flexible architectures.

References

1. Abraham, A., Nath, B.: Evolutionary design of neuro-fuzzy systems—a generic framework. In: Proceedings of the 4th Japan-Australia Joint Workshop on Intelligent and Evolutionary Systems, pp. 106–113 (2000)
2. Ahn, C.W., Ramakrishna, R.S.: Elitism-based compact genetic algorithms. IEEE Trans. Evol. Comput. **7**(4), 367–385 (2003)
3. Cordón, O.: A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: designing interpretable genetic fuzzy systems. Int. J. Approx. Reason. **52**(6), 894–913 (2011)
4. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2011)
5. Frenzel, J.F.: Genetic algorithm. IEEE Potentials **12**(3), 21–24 (1993)
6. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. IEEE Trans. Evol. Comput. **3**(4), 287–297 (1999)
7. Huang, P., Huang, Y., Wang, W., Wang, L.: Deep embedding network for clustering. In: 22nd International Conference on Pattern Recognition, Sweden, 24–28 Aug, pp. 1532–1537 (2014)
8. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)
9. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**(7553), 436–444 (2015)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Haykin, S., Kosko, B. (eds.) Intelligent Signal Processing, vol. 86, pp. 306–351. IEEE Press (2001)
11. LeCun, Y., Cortes, C., Burges, C.J.C.: The MNIST database for handwritten digits. http://yann.lecun.com/exdb/mnist/
12. Li, F., Qiao, H., Zhang, B., Xi, X.: Discriminatively boosted image clustering with fully convolutional auto-encoders (2017) (Online). https://arxiv.org/abs/1703.07980
13. Mininno, E., Cupertino, F., Naso, D.: Real-valued compact genetic algorithms for embedded microcontroller optimization. IEEE Trans. Evol. Comput. **12**(2), 203–219 (2008)

14. Mininno, E., Neri, F., Cupertino, F., Naso, D.: Compact differential evolution. IEEE Trans. Evol. Comput. **15**(1), 32–54 (2011)
15. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (Coil-20). Technical Report No. CUCS-006-96, Department of Computer Science, Columbia University, New York, NY, United States (1996)
16. Paul, S., Kumar, S., Singh, L.: Novel hybrid compact genetic algorithm for simultaneous structure and parameter learning of neural networks. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, Australia, 10–15 June, 2012. pp. 1–6 (2012)
17. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm Intell. **1**(1), 33–57 (2007)
18. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers (2017) (Online). https://arxiv.org/abs/1703.01041
19. Rere, L.M.R., Fanany, M.I., Arymurthy, A.M.: Metaheuristic algorithms for convolution neural network (2016) (Online). https://arxiv.org/abs/1610.01925
20. Salimans, T., Ho, J., Chen, X., Sutskever, I.: Evolution strategies as a scalable alternative to reinforcement learning (2017) (Online). https://arxiv.org/abs/1703.03864
21. Singh, L., Kumar, S., Paul, S.: Automatic simultaneous architecture and parameter search in fuzzy neural network learning using novel variable length crossover differential evolution. In: Proceedings of FUZZ-IEEE 2008, IEEE International Conference on Fuzzy Systems, pp. 1795–1802 (2008)
22. Wang, Y., Xu, C., Qiu, J., Xu, C., Tao, D.: Towards evolutional compression (2017) (Online). https://arxiv.org/abs/1707.08005
23. Xie, L., Yuille, A.: Genetic CNN (2017) (Online). https://arxiv.org/abs/1703.01513

Part II
Intelligent Social Informatics

Hybrid Approach for Predicting and Recommending Links in Social Networks

Shashi Prakash Tripathi, Rahul Kumar Yadav, Abhay Kumar Rai and Rajiv Ranjan Tewari

Abstract The prevalence of social networks like Myspace, Facebook, Hi5, and LinkedIn is increasing day by day, as they provide a platform where users of the social network can share their content. They also facilitate their users by recommending new friends on the basis of local or global network features. Local feature-based approaches do not exploit the whole network structure. In contrary to the techniques based on local features, global feature-based techniques make use of the complete network structure, being less efficient for large social networks. Here, we define a hybrid feature-based approach that uses local graph feature by computing proximity between every pair of nodes. It also captures global feature by computing all length two and length three pathways between each pair of vertices of the network. We performed experimental evaluation by comparing the proposed approach with other friend recommendation techniques. The experimental results indicate that our algorithm provides adequate level of efficiency as well as accuracy in friend recommendations.

Keywords Online social networks · Friend recommendation · Link prediction

1 Introduction

Social networks like Hi5, Myspace, and LinkedIn contain tremendous amount of data that can be used to predict missing links for friend recommendations. Social network-

S. P. Tripathi · R. K. Yadav · A. K. Rai (✉) · R. R. Tewari
Centre of Computer Education, IPS, University of Allahabad, Allahabad 211002, India
e-mail: abhay.jk87@gmail.com

S. P. Tripathi
e-mail: itsshavar@outlook.com

R. K. Yadav
e-mail: rahulkryadav93@gmail.com

R. R. Tewari
e-mail: tewari.rr@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_9

ing systems collect useful information from social contacts of their users, construct an interconnected graph using collected information, and recommend friends to other members of the graph on the basis of some similarity measure. The friend recommendations are beneficial for the users of a social network because sometimes an individual might be unaware of the presence of a desirable friend who is only at a few steps from him. So, users of a social network may expand their own social circle through these friend recommendations.

In a social network, a group of users can share informative items such as news, photos, and articles. They can also agree to be listed as friends to each other. Even, individuals can form links with other members of the network using various ways. Any individual in the network may also form a new group and ask others to join the group. On receiving the request initiated by any one among the existing users, other person can accept or reject it.

The proposed approach uses two similarity scores for friend recommendations. Each one of these is based on local as well as global features of the network. Final similarity scores are calculated from a formula which uses these hybrid feature-based approaches. The final similarity scores are used for recommending friends in the network.

1.1 Motivation

Some popular social networks like Hi5 and Facebook, as in Fig. 1, have used only local feature of the graph for new friend recommendations to a particular person v_5: "People you may know: (i) users v_1, v_3 because you have one common friend v_2, and (ii) user v_6 because you have one common friend v_{11}." So, for recommending friends

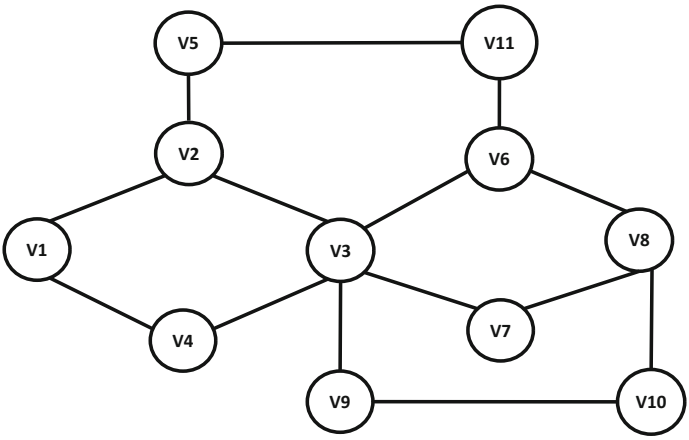


Fig. 1 Example of a social network

they take maximum length-two pathways only. Here, in this example we cannot rank friend recommendations for v_5 because v_5 gets friend recommendations with respect to v_1, v_3, and v_6 with equal probability, as the number of common neighbors are same for all recommended friends.

However, v_6 should have higher probability for friend recommendation to v_5 as compared to v_1 and v_3 because there exist a "strong" connection between v_5 and v_{11}, as v_{11} does not share many edges with others while v_1, v_3 should have lower probability because of the "loose" connection between v_5 and v_2, as v_2 shares more links with other vertices.

Compared to other existing techniques, the proposed approach estimates local as well as global graph features while computing similarity values corresponding to nodes present in the network. Firstly, we define a local feature-based basic similarity function to estimate the closeness between each vertex pair of the graph. Further, we capture two global features of the chosen graph, one based on transitive node similarity and second one based on different path lengths between the pair of vertices of a social network.

1.2 Contribution

Main outcomes of the proposed method are enumerated as follows: (i) A hybrid approach is designed for recommending friends in social networks. (ii) The designed method gives more accurate friend recommendations, using local as well as global features of the graph. (iii) A transitive node similarity measure is used that considers the shortest route between the selected vertex pair in a social network. (iv) We provide higher efficiency by limiting our traversal to 2 and 3 length paths only in a network. (v) To calculate the final similarity scores, we define a formula which uses two hybrid feature-based approaches. (vi) We carry out our experimental evaluation by comparing the proposed approach with other friend recommendation techniques on two data sets, namely Karate Club and Jazz Club. (vii) The proposed method substantially improves accuracy as well as efficiency of recommending friends corresponding to other existing methods, as will be shown experimentally.

Following this introductory section, Sect. 2 recapitulates the works carried out by others in the related area. Section 3 gives a description of basic graph terminology and proposed method. Section 4 contains detailed experimental analysis. Finally, last section concludes outcomes of the work done and summarizes main findings of the proposed work.

2 Related Work

In light of his "small-world" hypothesis, Stanley Milgram reported that "everyone in the world could be connected to everyone else via six degree of separation" [1].

According to homophily theory [2], "individuals tend to prefer the same things that similar other users do like." Watts et al. [3] experimented with the "algorithmic small-world hypothesis," and affirmed, "half of all existing chains can be completed in six to seven steps, supporting six degrees of separation hypothesis." The link prediction research area in network analysis focuses on finding which new relationships among users of an online social network will have higher probability to exist in future. The said problem can be tackled by two main approaches [4]. First one utilizes local graph features, while another one utilizes global graph features by exploiting complete graph structure.

A variety of local feature-based approaches [4] like "Common Neighbors index," "Adamic/Adar index," "Jaccard Coefficient" are available in the literature, for examining the closeness between a particular pair of vertices of a graph. Out of all these measures, "Adamic/Adar" [5] reported to achieve the adequate level of performance in predicting missing links in the graph corresponding to a social network. Some popular social networks like Hi5 and Facebook use "Common Neighbors index, also known as friend of a friend (FOAF)" [6] technique for recommending friends. In FOAF, if a vertex pair v_x and v_y has a large number of common adjacent vertices, then they will have more probability to establish a new edge in future. Some other local feature-based techniques are also available in the literature which are based on the "preferential attachment" [4].

A variety of global feature-based approaches [4] like "Random Walk with Restart (RWR)," "SimRank," and "Shortest Path" are also available in the literature. Liben-Nowell and Kleinberg [4] used shortest path technique for predicting friend recommendations between vertex pairs of a graph. Any well-known technique for finding shortest path [7, 8] might be applied for finding the shortest path between a particular vertex pair. RWR [9] method is founded on the widely known "Markov chain model of random walk." SimRank [10, 11] uses a global graph feature which depends on the structural attributes of the graph which infers "two objects are similar if they are related to similar objects."

A variety of hybrid feature-based approaches like FriendLink [12], FriendTNS [13] are also available in the literature which are based on local as well as global features of the network. FriendLink [12] uses two main approaches for computing similarity scores. The first one is based on local graph feature, focussing on the node's structure and another one utilizes global graph feature, considering the whole path structure of the graph. FriendTNS [13] also uses two main approaches for computing similarity scores. The first one captures local graph feature by computing proximity between every pair of nodes. The second one uses global graph feature by weighing paths through which a particular node pair is connected with each other.

Our method is based on the hybrid approach which uses local as well as global features in computation of similarity scores. The novelty of the proposed method compared to other existing methods is enumerated as follows: (i) It counts only 2 and 3 path lengths in a graph so it can be considered as a local feature-based approach. The proposed method attains higher accuracy in recommending friends against other local feature-based techniques. The obvious justification is that we are expanding the user's neighborhood, thereby capturing more information. (ii) The

proposed approach is more efficient as compared to other global feature-based methods because we are not considering all path lengths while computing the similarity scores because if we consider path lengths of value more than 3, then it will not contribute significantly in final similarity scores, as similarity score values come after more than four decimal places. Even, it is more efficient than hybrid feature-based approaches like FriendLink and FriendTNS. The proposed algorithm attains more accuracy as well as efficiency than other existing methods which we have demonstrated experimentally in Sect. 4.

3 The Proposed Method

Suppose, to start with, a particular node is chosen for recommending friends. First, we quantify the similarity scores from this designated vertex to rest of the graph vertices using some predefined similarity measure that will be defined in Sect. 3.2. Finally, based on the calculated similarity scores, top-k vertices will be recommended as the friends to the designated vertex.

3.1 Basic Graph Terminology

Here, we consider a general undirected and unweighted graph $G = (V, E)$, where V stands for set of vertices and E stands for set of edges. An edge represents association betwixt a vertex pair. The graph G shown in Fig. 1 indicates overall interactions between members of a social networking system. The social network presented in Fig. 1 will be further used to run an example in outline section.

We can represent graph G in the form of an adjacency matrix A with rows and columns marked as the graph nodes, with either value 1 or 0 at location (v_i, v_j) depending on the friendship between vertex pair v_i and v_j. Here, symmetric adjacency matrix will be constructed because we have taken an undirected graph.

3.2 Defining Node Similarity Measures

Here, we define some similarity measures which will be used to express the proximity between the node pairs of a graph. Suppose v_i and v_j are two nodes of a graph and a function $sim(v_i, v_j)$ expresses similarity between node pair v_i and v_j. Greater similarity value between a vertex pair indicates higher possibility between them to become friend in future.

A person in a social network may be connected to some other person by many possible routes. We can define a similarity measure based on finding all possible routes of different lengths between each vertex pair of the graph. Some similarity

definitions assume that nodes in a social network may use all or some out of all pathways connecting them. Thus, a pair of persons connected via many distinct routes will have a higher chance to form an edge between them. It is also proportional to the length of pathways through which the persons are connected. In Fig. 1, if we choose paths with length two only, then v_3 would have equal probability of friend recommendations with nodes v_1 and v_8. But, if we consider paths of length three as well, v_3 will have higher possibility of friend recommendations with v_8 in contrast to v_1.

Definition 1 As defined in [12], if we consider all the length-l paths for computing similarity measure, then $sim(v_i, v_j)$ between any pair of nodes v_i and v_j is given as:

$$sim(v_i, v_j) = \sum_{k=2}^{l} \frac{1}{k-1} . \frac{\mid Paths^k(v_i, v_j) \mid}{\prod_{p=2}^{k}(n-p)}$$

where n represents vertex count in graph G, l stands for maximum path length considered in computing the similarity score between each vertex pair (only simple path without cycles is taken into consideration), $\frac{1}{k-1}$ is the "attenuation factor" for providing weights to path on the basis of length, $\mid Paths^k(v_i, v_j) \mid$ is the "number of k-length paths between v_i and v_j," and $\prod_{p=2}^{k}(n-p)$ is the "number of k-length paths between v_i and v_j, if G is a complete graph."

Definition 2 As defined in [13], the basic similarity measure $bSim(v_i, v_j)$ between a vertex pair v_i and v_j of a graph G is defined as:

$$bSim(v_i, v_j) = \begin{cases} 1, & if\ v_i = v_j \\ 0, & if\ v_i \neq v_j\ and\ (v_i, v_j) \notin E \\ & and\ (v_j, v_i) \notin E \\ \frac{1}{degree(v_i)+degree(v_j)-1}, & otherwise \end{cases}$$

where $degree(v_i)$ is the degree of ith node in G.

Definition 3 The proposed similarity measure $cSim(v_i, v_j)$ is defined as:

$$cSim(v_i, v_j) = [sim(v_i, v_j)\ .\ eSim(v_i, v_j)]^{\frac{1}{2}}$$

where $eSim(vi, vj)$ is the extended similarity measure, defined as:

$$eSim(v_i, v_j) = \begin{cases} 0, & if\ there\ is\ no\ path\ between\ v_i \\ & and\ v_j \\ bSim(v_i, v_j), & if\ v_i\ and\ v_j\ are\ adjacent \\ \prod_{h=1}^{k} bSim(v_{p_h}, v_{p_{h+1}}), & otherwise \end{cases}$$

where $v_{p_1} = v_i$, $v_{p_{k+1}} = v_j$, and vertices v_{p_h}(for $h = 2, 3, 4, \ldots, k$) are the intermediate vertices through which the shortest path between v_i and v_j passes.

3.3 Outline

Here, firstly we discuss the proposed approach FriendREC through a running example. Our FriendREC approach computes similarity scores between each vertex pair of a graph. It takes any connected graph G as the input and gives a similarity matrix as the output. A particular entry in the similarity matrix represents similarity score corresponding to a particular vertex pair of graph G. Now, we can recommend friends to a target user using similarity scores between target user and rest of the nodes. To understand working of the proposed method, we can run the FriendREC algorithm on an input graph. Eleven persons are connected in the graph shown in Fig. 1. Suppose we wish to suggest a new friend to node v_3. First of all we preprocess given data set to find all existing paths of length two and length three among v_3 and all other nodes of the graph. Table 1 shows matrices containing all length two and length three paths between every pair of vertices of graph G.

Our FriendREC approach uses the formula given in Definition 3 of Sect. 3.2. The given formula consists of two parts. To get the final similarity score between any pair of nodes, we compute both the parts separately and then we take the geometric mean of these values. Let us take an example, to illustrate the computation of similarity scores between the node pairs v_3, v_1 and v_3, v_8.

The scores corresponding to first part, $sim(v_i, v_j)$ between v_3, v_1 and v_3, v_8 are calculated on the basis of 2 and 3 length paths which connect them. Possible paths from v_3 to v_1 are $(v_3 \rightarrow v_2 \rightarrow v_1)$ and $(v_3 \rightarrow v_4 \rightarrow v_1)$, and possible paths between v_3 and v_8 are $(v_3 \rightarrow v_6 \rightarrow v_8)$, $(v_3 \rightarrow v_7 \rightarrow v_8)$, and $(v_3 \rightarrow v_9 \rightarrow v_{10} \rightarrow v_8)$. Each two length path has a weight, i.e., 0.1111 whereas three length path has a weight equal to 0.0069. Therefore, $sim(v_3, v_1) = 0.1111 + 0.1111 = 0.2222$ and $sim(v_3, v_8) = 0.1111 + 0.1111 + 0.0069 = 0.2291$. The scores corresponding to second part, $eSim(v_i, v_j)$ between v_3, v_1 and v_3, v_8 make use of shortest path and basic similarity $bSim(v_i, v_j)$ values between them. Shortest path between v_3 and v_1 is $(v_3 \rightarrow v_2 \rightarrow v_1)$ then $eSim(v_3, v_1) = bSim(v_3, v_2) * bSim(v_2, v_1) = 0.1428 * 0.2500 = 0.0357$. Similarly, shortest path between v_3 and v_8 is $(v_3 \rightarrow v_6 \rightarrow v_8)$ then $eSim(v_3, v_8) = bSim(v_3, v_6) * bSim(v_6, v_8) = 0.1428 * 0.2000 = 0.0286$.

Now calculating final similarity scores between v_3, v_1 and v_3, v_8 by FriendREC approach, we get, $cSim(v_3, v_1) = (0.2222 * 0.0357)^{\frac{1}{2}} = 0.0890$. Similarly, $cSim(v_3, v_8) = (0.2291 * 0.0286)^{\frac{1}{2}} = 0.0809$. So, we can say that v_1 is more close to v_3 as compared to v_8. Therefore, v_1 would be recommended as the friend to v_3. So, the accuracy of friend recommendations can be improved by considering associations between different pair of nodes. The steps included in our FriendREC algorithm are shown in Algorithms 1 and 2.

Algorithm 1 FriendREC (G, Adj, N, l)

Input: G stands for input graph, Adj stands for the adjacency matrix, N stands for vertex count, and l stands for the maximum path length being considered.

Output: Combined similarity measure $cSim(v_i, v_j)\ \forall\ v_i$ and v_j in G.

```
Begin
sp ← list(), eSim ← matrix(data = 1, nrow = N, ncol = N)
for i ← 1 to N do
  for j ← 1 to N do
    sp ← Calculate shortest path between i^th and j^th node
    for k ← 1 to [length(sp) − 1] do
      eSim[i, j] ← eSim(i, j) * bSim[k, k + 1]
    end for
  end for
end for
for r ← 2 to l do
  sim ← Similarity(sim, r)
end for
for i ← 1 to N do
  for j ← i + 1 to N do
    cSim[i, j] ← sqrt(Sim[i, j] * eSim[i, j]
    cSim[j, i] ← cSim[i, j]
  end for
end for
End FriendREC
```

Algorithm 2 Similarity (sim, r)

Input: sim is the input similarity matrix, and r is the path length for which we have to compute similarity score.

Output: Computed similarity matrix sim.

```
Begin
for i ← 1 to N do
  for j ← 1 to N do
    den ← 1
    for k ← 2 to r do
      den ← den * (N − k)
    end for
    sim[i, j] ← sim[i, j] + 1/(r−1) * |paths_r(i, j)| / den
  end for
end for
return sim
End Similarity
```

Table 1 Matrices representing all the paths of (a) length two, and (b) length three of the graph in Fig. 1

(*a*) *Length two*

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
v_1	0	0	2	0	1	0	0	0	0	0	0
v_2	0	0	0	2	0	1	1	0	1	0	1
v_3	2	0	0	0	1	0	0	2	0	1	1
v_4	0	2	0	0	0	1	1	0	1	0	0
v_5	1	0	1	0	0	1	0	0	0	0	0
v_6	0	1	0	1	1	0	2	0	1	1	0
v_7	0	1	0	1	0	2	0	0	1	1	0
v_8	0	0	2	0	0	0	0	0	1	0	1
v_9	0	1	0	1	0	1	1	1	0	0	0
v_{10}	0	0	1	0	0	1	1	0	0	0	0
v_{11}	0	1	1	0	0	0	0	1	0	0	0

(*b*) *Length three*

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
v_1	0	1	0	1	0	2	2	0	2	0	1
v_2	1	0	1	0	0	1	0	2	0	1	1
v_3	0	1	0	1	1	1	1	1	0	2	1
v_4	1	0	1	0	2	0	0	2	0	1	1
v_5	0	0	1	2	0	1	1	1	1	0	0
v_6	2	1	1	0	1	0	0	1	1	1	0
v_7	2	0	1	0	1	0	0	1	1	1	2
v_8	0	2	1	2	1	1	1	0	2	0	0
v_9	2	0	0	0	1	1	1	2	0	0	1
v_{10}	0	1	2	1	0	1	1	0	0	0	1
v_{11}	1	1	1	1	0	0	2	0	1	1	0

4 Experimental Evaluation

Here, we experimentally evaluate our FriendREC algorithm against six other existing link prediction techniques. In particular, we compare it with the FriendLink [12], the FriendTNS [13], the FOAF [6], the Adamic and Adar [5], the Katz [14], and the PA [15]. All algorithms are implemented in R programming language.

4.1 Evaluation Metrics and Data Sets

The metrics used in our evaluation are as follows: Precision, recall, and accuracy measures are used to quantify the performance of friend recommendation algorithms. For time comparison, we use efficiency as the performance measure. **Precision** is the "fraction of relevant persons among the predicted persons." **Recall** is the "fraction of relevant persons that have been predicted among the total relevant persons." Furthermore, **AUC** (area under the curve) statistic is used to assess the accuracy of a particular friend recommendation technique. It is determined by "area under the receiver-operating characteristic (ROC) curve."

In our evaluation, we consider two categories of data sets. The first category is of real data sets and second one is of synthetic data sets. For measuring accuracy, precision, and recall, we use two real data sets. For efficiency analysis, we use synthetic data sets.

4.2 Accuracy Comparison of FriendREC with Other Methods

Here, FriendREC method is compared with FriendTNS, FriendLink, FOAF, and Adamic/Adar techniques with accuracy, precision, and recall as the evaluation metrics using two real data sets, namely Karate Club [16] having 34 nodes, and Jazz Club [17] having 198 nodes.

The histograms shown in Fig. 2 quantify the soundness of different methods in achieving high recall value with minimum loss in precision value. Our proposed algorithm FriendREC produces the best result with high precision value in comparison with all other existing techniques. The obvious rationale is that FriendREC uses global as well as local features of the graph.

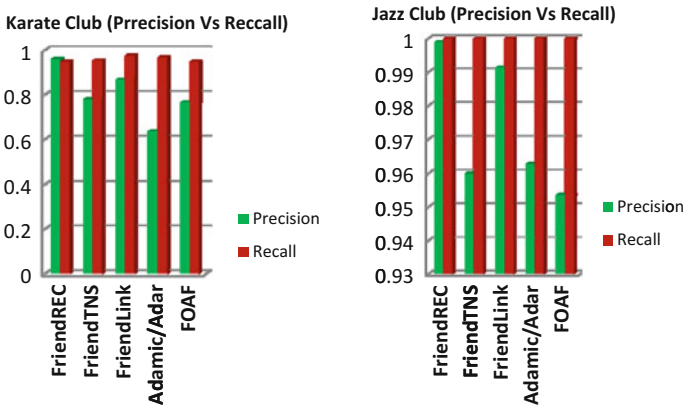


Fig. 2 Comparison of FriendREC with other existing techniques using Karate Club and Jazz Club

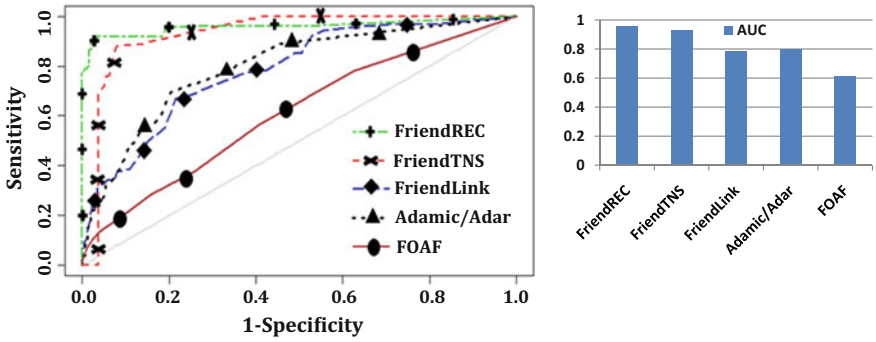


Fig. 3 Comparison of ROC and its corresponding AUC of FriendREC algorithm against other methods for Karate Club

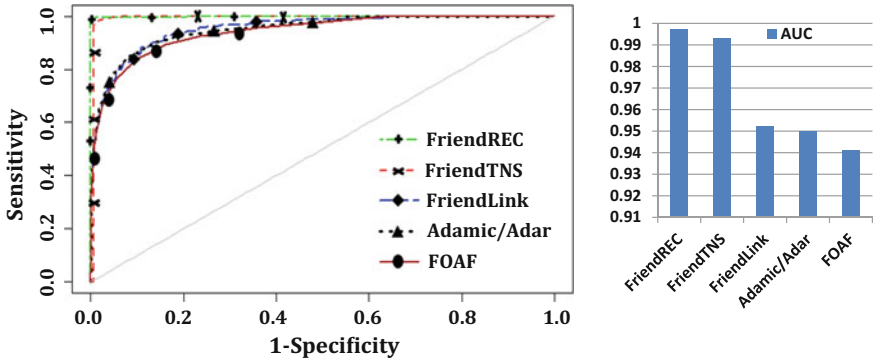


Fig. 4 Comparison of ROC and its corresponding AUC of FriendREC algorithm against other methods for Jazz Club

The graphs and corresponding histograms shown in Figs. 3 and 4 present the accuracy performance of FriendREC against other existing methods in terms of AUC. The results clearly indicate that the FriendREC algorithm outperforms all other existing techniques in terms of AUC and hence accuracy.

4.3 Time Comparison of FriendREC with Other Methods

Here, FriendREC approach is compared with FriendTNS, FriendLink, PA, Katz, and Adamic/Adar techniques on the basis of time consumption using synthetic data sets having 500, 1000, 2000, 5000, and 10,000 nodes. We have measured the computing time for all algorithms considering their off-line parts. The computed results are given in Table 2. According to Table 2, FriendREC outperforms FriendLink, FriendTNS, PA, Katz, and Adamic/Adar, since they require more computation. Here, it is

Table 2 Computing time (in seconds) of the synthetic data sets

Data-Set(N,k)	FriendREC	FriendTNS	FriendLink	PA	Katz	Adamic/Adar
Synthetic (500, 5)	0.0002	0.004	0.001	0.002	0.061	0.002
Synthetic (1000, 10)	0.0050	0.050	0.010	0.020	0.130	0.024
Synthetic (2000, 20)	0.0130	0.090	0.050	0.030	0.490	0.050
Synthetic (5000, 50)	0.0680	0.180	0.100	0.410	2.720	0.240
Synthetic (10000, 100)	0.2100	1.170	0.560	0.546	15.88	0.369

important to note that FriendREC outperforms Adamic/Adar which is a local feature-based similarity measure having lower complexity than global feature-based measures.

The computed values in Table 2 represent the total time needed to calculate the complete similarity matrix. However, if we have to find the similarity values corresponding to a single user, the calculation will take very little amount of time.

5 Conclusion

The importance of social networking systems is increasing day by day because they provide a common platform to share contents like articles, audios, videos, and photos. The users can also expand their social circle by establishing new interactions with other members of the network. A new hybrid feature-based approach is introduced and implemented in this paper which provides more precise friend suggestions. The advantages of the proposed approach over other existing approaches are as follows: (i) A new hybrid feature-based similarity function has been defined. The proposed function takes into account only length two and length three paths that connect a user to other users in a social network, based on "algorithmic small-world hypothesis." (ii) We have accomplished experimental evaluation of the FriendREC approach against six representative friend suggestion techniques using real and synthetic data sets. The experimental outcomes clearly indicate that the FriendREC method gives more accurate friend suggestions as compared to other existing techniques, as shown experimentally in Sect. 4.2. (iii) The proposed method is more efficient than other existing techniques, as shown experimentally in Sect. 4.3.

References

1. Travers, J., Milgram, S.: The small world problem. Psychol. Today **1**, 61–67 (1967)
2. Bisgin, H., Agarwal, N., Xu, X.: A study of homophily on social media. World Wide Web **15**(2), 213–232 (2012)
3. Goel, S., Muhamad, R., Watts, D: Social search in small-world experiments. In Proceedings of the 18th international conference on World Wide Web, pp. 701–710. ACM (April 2009)
4. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Assoc. Inf. Sci. Technol. **58**(7), 1019–1031 (2007)
5. Adamic, L., Adar, E.: How to search a social network. Soc. Netw. **27**(3), 187–203 (2005)
6. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make new friends, but keep the old: recommending people on social networking sites. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 201–210. ACM (April 2009)
7. Thomas H..Cormen, Leiserson, C. E., Rivest, R. L., Stein, C.: Introduction to algorithms (Vol. 6). Cambridge: MIT press (2001)
8. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. J. ACM (JACM) **34**(3), 596–615 (1987)
9. Pan, J. Y., Yang, H. J., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 653–658. ACM (August 2004)
10. Jeh, G., Widom, J.: SimRank: a measure of structural-context similarity. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 538–543. ACM (July 2002)
11. Yu, W., Lin, X., Le, J.: A space and time efficient algorithm for SimRank computation. In: 2010 12th International Asia-Pacific on Web Conference (APWEB), pp. 164–170. IEEE (April 2010)
12. Papadimitriou, A., Symeonidis, P., Manolopoulos, Y.: Fast and accurate link prediction in social networking systems. J. Syst. Softw. **85**(9), 2119–2132 (2012)
13. Symeonidis, P., Tiakas, E.: Transitive node similarity: predicting and recommending links in signed social networks. World Wide Web **17**(4), 743–776 (2014)
14. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
15. Newman, M.E.: Clustering and preferential attachment in growing networks. Phys. Rev. E **64**(2), 025102 (2001)
16. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**(4), 452–473 (1977)
17. Gleiser, P.M., Danon, L.: Community structure in jazz. Adv. Complex Syst. **6**(04), 565–573 (2003)

Online Partitioning of Large Graphs for Improving Scalability in Recommender Systems

Vinit Koshti, NVS Abhilash, Karanjit S. Gill, Nikhil Nair, Mahima B. Christian and Prosenjit Gupta

Abstract All around the world people are using social media to create large online social networks, and the sizes of these networks are continuously increasing. It is estimated that by the year 2018, 2.44 billion people will be using social networks. The growing desire and demand of social networks are not just removing our communication barrier, and it is also changing the way we think and do things. The people we 'trust' or 'are close to' in our social network continuously shape our views about different things. This trust-based relationship can be harnessed to recommend items to users. In this paper, we aim to tackle the problem of scalability in online recommender systems using an online graph partitioning algorithm.

Keywords Recommender systems · Online social networks · Trust computation Online algorithms

V. Koshti (✉) · N. Abhilash · K. S. Gill · N. Nair · M. B. Christian · P. Gupta
Department of Computer Science and Engineering, NIIT University, Neemrana, Rajasthan, India
e-mail: vinitkoshti.95.vk@gmail.com

N. Abhilash
e-mail: nvs232@gmail.com

K. S. Gill
e-mail: ksg207@gmail.com

N. Nair
e-mail: nnair25@gmail.com

M. B. Christian
e-mail: mahimachristian96@gmail.com

P. Gupta
e-mail: prosenjit_gupta@acm.org

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_10

1 Introduction

1.1 Collaborative Filtering

Collaborative filtering is a popular technique used in recommender systems which make use of the 'wisdom of the crowd' to find correlations between users and items to recommend top-k items to a target user. Over the last decade or so, recommender systems have become popular choices to help users tackle the information overload and choose the items they would like. At the same time, today increasingly a large number of people are joining online social networks (OSNs). Tobler's First Law of Geography states that 'Everything is related to everything else, but near things are more related than distant things.' This applies to users and their choices as well. People are more influenced by others who are close to them. The popularity of OSN's has turned the notion of proximity on its head. The notion of 'six degrees of separation' proposed by the Hungarian author and journalist Frigyes Karinthy in 1929 assumes that everyone in this world is six or fewer steps away from anyone else via friendship relationships. In February 2016, Facebook Research reported that at least among the 1.59 billion Facebook users, the average degree of separation has actually shrunk to 3.57. As a result, people are increasingly influenced in their choices by others who are close to them on OSNs.

From the above discussions, it is clear that the social network graphs contain useful information that can be harnessed for the purpose of recommendations. However, the large size of such graphs present significant challenges. We can partition the graph and process the partitions independently.

Whereas this can improve scalability, partitioning has to be done carefully to prevent loss of accuracy in the recommendation process. To ensure this, we will be *using Condensed Spanning Tree* data structure and *Online Algorithms* together to form coherent partitions.

1.2 Social Trust

Under their computational model, Jøsang et al. [1] define trust as

> The subjective probability by which one user expects that another user performs a given action.

Described below are some closely related concepts to trust:

1. Social influence: Suppose A wants to know the trustworthiness of C and A asks B for advice. If B suggests something to A, and A decides to do it, we can say that A is influenced by B.
2. Recommendation is the method that allows opinions to propagate. During the process of trust evaluation from A to C, the trust from A to B can be taken as

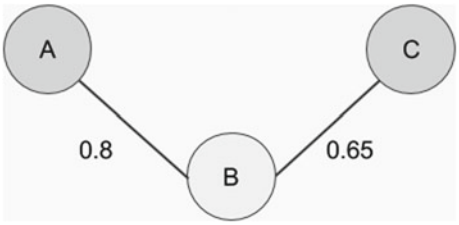


Fig. 1 Users A, B, C in social network with edge weight's quantifying trust

referral trust as in Jøsang et al. [2], which is trust with respect to the ability to recommend a good target. *B* can be taken as recommenders. The final trust that *A* puts in *C*, through the referrals of *B*, is called recommendation trust [3], which is trust derived from others' experiences (Fig. 1).

Our new trust evaluation formula is based on the definition of trust by Josang. Additionally, we have also developed a derived trust heuristic for increasing the cohesiveness of the partitions/clusters.

1.3 Condensed Spanning Tree

We use the *Condensed Spanning Tree* (CST) structure of Filippidou and Kotidis [4]. Considering the graph with nodes and edge weights between them, in a Condensed Spanning Tree, each node u has a weight $S(u)$, where $S(u)$ is the sum of weights of nodes in the subtree rooted at node u. The parent edge of each node u has a weight equal to $E(u)$, where E is the sum of weights of the edges that emanate from the subtree to a node not in the subtree. Also, there are two methods to propagate across the graph, (a) node weight propagation—to update size of subtree and (b) edge weight propagation—to update edge-cut cost.

Formation of CST. In context of a streaming model where edges arrive in arbitrary order. There are three cases:

1. If both nodes u and v do not appear in the tree
2. If one is present and another is not present.
3. If both u and v are present in tree.

   a. If u and v in same cluster
   b. If u and v in different cluster.

The CST algorithm is varied according to each case.

We have used CST creation algorithm as proposed in [4]. After creating the CST, we need to partition the graph. The partitioning algorithm is as follows

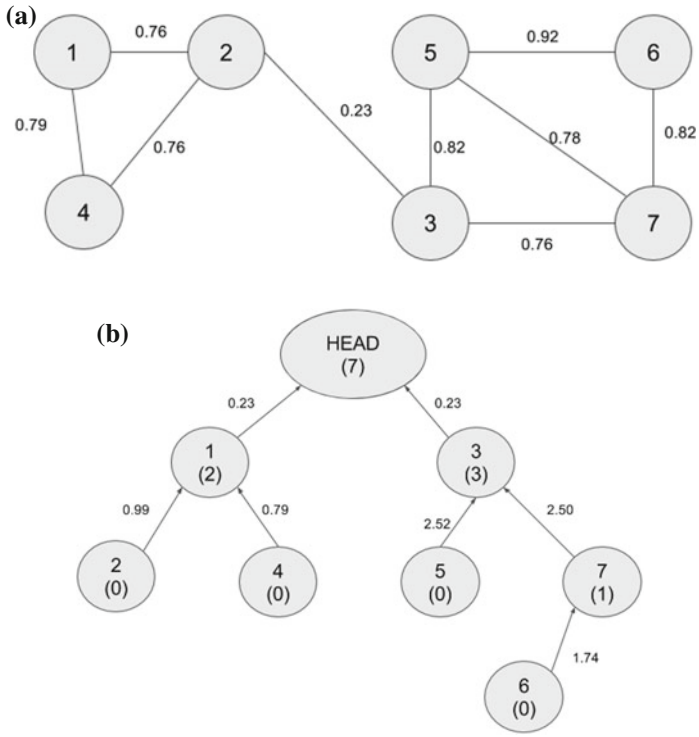


Fig. 2 **a** Sample social network, **b** Generated condensed spanning tree of sample social network

On-Demand Partitioning. The graph partitioning algorithm is transformed into a multiple knapsack problem by considering nodes as 'items.'

The edge weight of each node *E* corresponds to the edge-cut cost.

The subtree weight *S* corresponds to the size of each item.

$$\text{Size of each partition/knapsack} = |\text{V}|/\text{k}. \tag{1}$$

$$\text{Amortized Cost} = \text{E/S} = \text{edge} - \text{cut cost/size of the subtree}. \tag{2}$$

The algorithm sorts nodes in ascending order based on the amortized cost.

1. Select nodes from the list that do not violate the balance (bin capacity) criterion.
2. When $k-1$ partitions have been formed, assign the remaining nodes to the last, kth partition.

Now here we present an example to demonstrate the use of Condensed Spanning Tree in our work. Let us say that we have the following graph (Fig. 2).

The weights in the graph are the edge weights that can be calculated using our proposed method in the following section. Now when we apply the CST creation algorithm to this graph, we get the following CST (Fig. 3).

Now after partitioning the graph, we get the following two partitions (K (the number of partition) was set to 2 here for simplicity).

Partitions:

Partition 1: [5–7]

Partition 2: [4, 8–10]

The above partitioning occurred because of the weak edge between Node 2 and Node 3. From this example, we can show that if we can make the edges between close friends in a social network stronger than other edges, we can use CST or other graph partitioning algorithms to partition the graph efficiently.

In Sect. 2, we discuss our contributions in partitioning of large graphs to improve scalability in recommender systems. First in Sect. 2.1, we propose the calculation of edge weights between two users in the social graph. Then using these edge weights generated, we create a CST. Then we run the CST partitioning algorithm to generate our seed clusters. Further to increase cohesiveness of each cluster, we add more edges in the graph using the derived trust heuristics that we have explain in Sect. 2.2. After that we use our online heuristics explained in Sect. 2.3 to add the new incoming edges to a specific partition on the go. In Sect. 3, we provide some of the observations and results based on our experiments. In Sect. 3.1, we describe the dataset on which the experiments were conducted. Then in Sect. 3.2, we share some statistics of the edge weight calculation algorithm. In Sect. 3.3, we discuss some of the key observations with the online heuristics and derived trust heuristic. It is important to note that our whole work tries to make coherent partitions of the social graph, so that each partition can be an input to any collaborative filtering algorithm to generate recommendations. We conclude in Sect. 4.

2 Our Contributions

2.1 Edge Weight Calculation

In this section, we propose an edge weight calculation method based on the location and social connection between the two users. Our formula is based on Josang's definition of trust. We have scaled our weights to [0, 1] where 0 represents **no trust** and 1 represents **full trust**.

Edge Weight Calculation Between Users. For this, we look at the common location visited by the users and the similarity between their ratings. The formula that we propose to calculate the edge weights between users is provided in Eq. (3). There W is the weight of the edge, CLR is the common location ratio between the two users, CoR is the correlation between the ratings of the common locations visited by the two users.

$$W = (CLR \times CoR) + FB. \tag{3}$$

Common Location Ratio. We first define the common location ratio between a given pair of users u_i and u_j and denote it as α (u_i, u_j). Let L_i and L_j denote the sets of locations visited by u_i and u_j, respectively. Then, the common location ratio, measuring the degree of overlap in visited locations, is defined as the Jaccard coefficient of L_i and L_j as shown in Eq. (4).

$$\alpha(u_i, u_j) = |L_i \cap L_j| / |L_i \cup L_j|. \tag{4}$$

Correlation between Users. Fundamentally, correlation states how similar are the views/choice of the two users, by comparing their views/choice at points of overlap. In our case, it is a measure to quantifies how similar are the ratings of the common location visited by the two users. Now we examine the ratings of the common locations between the two users ($L_i \cap L_j$). And we find S_i the number of locations having similar ratings and D_i the number of locations having dissimilar ratings. Then we define the correlation between the two users as the ratio of the difference of S_i and D_i to the total number of common location visited by the two users as shown in Eq. (5).

In the later sections of the paper, we use terms like positive correlation and negative correlation between the users. From Eq. (5), it is clear that we have positive correlation between two users when S_i is greater than D_i. And we have negative correlation when S_i is less than D_i.

$$CoR = (S_i - D_i) / |L_i \cap L_j|. \tag{5}$$

Friendship Bias. It is a constant value which denotes that u_i is a friend of u_j. It can be considered as a base trust between two friends.

Now using the above definitions and formulas, we present an algorithm to generate the edge weights for the users, given their visited locations and their corresponding rating vectors.

Note for the algorithm.

- Number of locations visited by user_1 = L1.
- Number of locations visited by user_2 = L2.
- dataStruct contains arrays of location IDs visited by User_1 and User_2 and their corresponding ratings, these arrays are sorted by location ID.

We are simultaneously moving in arrays of location IDs visited by User_1 and User_2.

If User 1 and User 2 have common location ID, then do stuff… Else if locIDofUser1 is less than locIDofUser2, move ahead in location array of user 1, if locIDofUser1 is greater than locIDofUser2, then move ahead in location array of user 2.

Now the algorithm is as follows.

Edge Weight Calculation

```
float edgeWeightCalculation (user_1, user_2, dataStruct )
  1:{
  2:     float uniqueLocationCnt = L1 + L2;
  3:     float commonLocationCnt = 0;
  4:     simChoiceCnt=0, nonSimChoiceCnt=0;
  5:     float CLR =0, CoR=0, weight=0.0;
  6:     for(i=0,j=0; i < L1 and j < L2; )
  7:     {
  8:
  9:         if(locationIdOfUser_1 == locationIdOfUser_2)
  10:        {
  11:             commonLocationCnt ++;
  12:             uniqueLocationCnt  - -;
  13:             if(rating by both users >=4)
  14:                 simChoiceCnt++;
  15:             else if(rating by both users < 4)
  16:                 simChoiceCnt++;
  17:             else
  18:                 nonSimChoiceCnt++;
  19:             i++;
  20:             j++;
  21:        }
  22:        else if (locationIdOfUser_2 > locationIdOfUser_1)
  23:             i++; //Move ahead in location array of user 1
  24:        else
  25:             j++; //Move ahead in location array of user 2
  26:    }
  27:    CLR = commonLocationCount/uniqueLocationCount;
  28:
  29:    // If CLR=0 then CoR will be undefined.
  30:    if(CLR != 0)
  31:     CoR=(simChoiceCnt-nonSimChoiceCnt)/commonLocationCnt;
  32:    weight = (CLR * CoR) + FRIEND_BIAS;
  33:
  34:    return weight;
  35:}
```

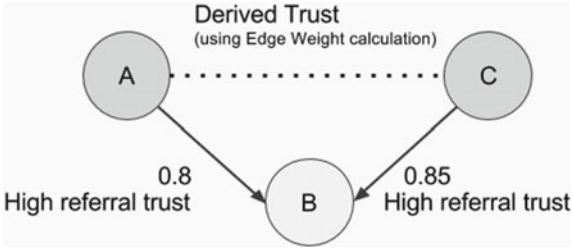


Fig. 3 Explaining derived trust using diagram

2.2 Derived Trust Heuristic

In this section, we discuss the advantages of using the derived trust heuristics to improve the cohesiveness of the individual partitions, which helps us to add edges for the hidden relationship between the users. This step would be very beneficial for any recommender system to generate better recommendations for related users.

2.2.1 Referral Trust and Derived Trust

Referral Trust. It is the trust in the ability of a recommender to recommend a good service provider.

Derived Trust. Suppose we have three users A, B and C, if A has high referral trust in B and if C has high referral trust in B then we define trust originated between A and C as *derived trust*.

Figure 3 describes the derived trust between A and C because of the referral trusts between A, B and C, B.

2.2.2 Derived Trust Heuristic in Our Work

Below is the Venn diagram where the universal set is the set of all geographical location that can be visited by the users.

L_A = set of locations visited by user A
L_B = set of locations visited by user B
L_C = set of locations visited by user C

Note that more the sets of locations of A, B, and C intersect implies they are more correlated according to Eq. (5) in Sect. (2.1).

Now we show that it is sensible to add an edge between A and C, i.e., A and C are also correlated, only when A and C both have a positive correlation with B.

Let us assume a corner case of positive correlation between A, B and C, B as shown in Fig. 4a. In this case, sets L_A, L_B, and L_C just touch each other. In such a

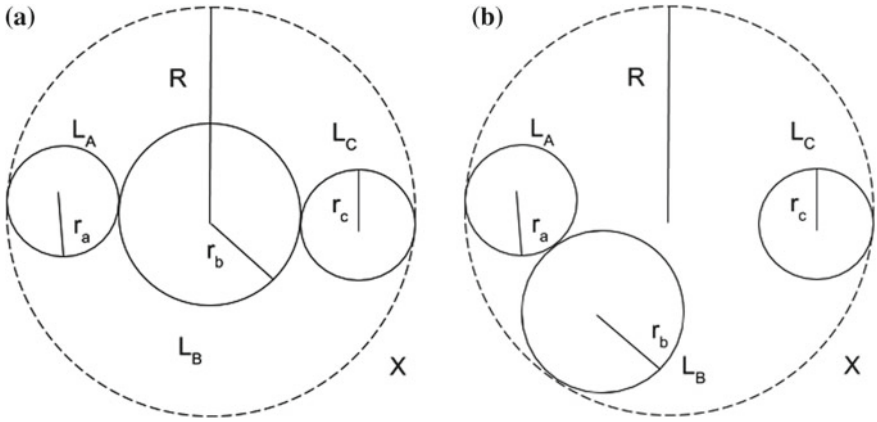


Fig. 4 Venn diagram location space (Geographical)

case, the probability that C intersects with A, i.e., A and C have a positive correlation denoted as P_{AC}, is

$$\mathrm{P_{AC}} = \text{Area of location space of user A/Area of circle X.} \tag{6}$$

$$\mathrm{P_{AC}} = \pi(\mathrm{r_a})^2/\pi(\mathrm{r_x})^2 \tag{7}$$

$$\mathrm{P_{AC}} = \mathrm{r_a^2}/\mathrm{r_x^2} \tag{8}$$

Here r_a, r_b, and r_c are the radii of L_A, L_B, and L_C, respectively. And radius of the circle X for this corner case is:

$$\mathrm{r_x} = \mathrm{r_a} + \mathrm{r_b} + \mathrm{r_c}. \tag{9}$$

Also, when A, B and A, C have more correlation, r_x will decrease which in turns implies that P_{AC} will increase and which shows that if positive correlation between A, B and C, B is high then the probability of positive correlation between A, C is also high.

Figure 4b shows that when the location space of users does not overlap, radius r_x is greater than $(r_a + r_b + r_c)$, which will decrease the value of P_{AC}.

Hence, the above explanation suggests that we should add an edge (denoting positive correlation between the two nodes) when they have high positive correlation between a common user. Following the definition of derived trust, derived weight heuristic is based on transitive relationship that is if A has positive correlation with B and B has positive correlation with C, then there is a higher chance of A and C having positive correlation.

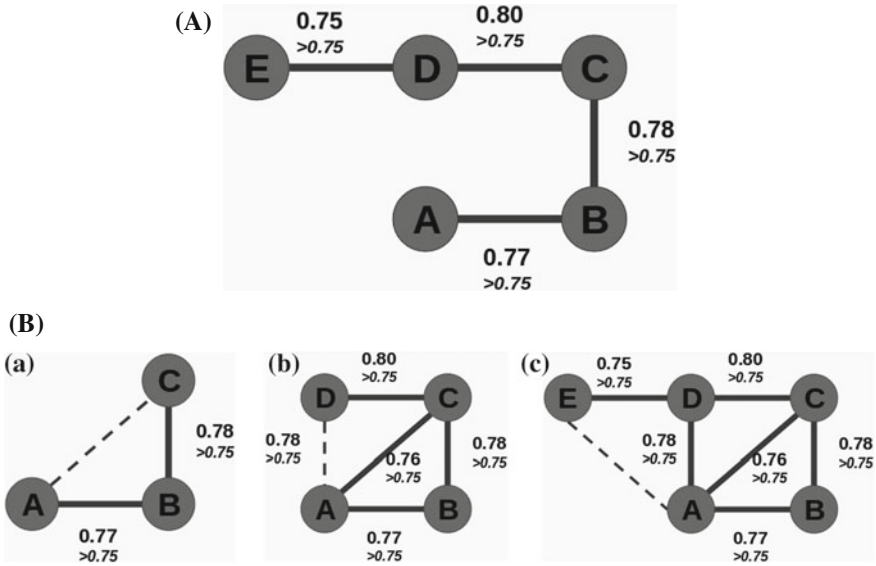


Fig. 5 **a** Sample social network, **b** Step-wise application of derived trust heuristic on sample social network

2.2.3 Advantages of Using Derived Trust Heuristic

Derived trust heuristic can be used to make the partitions generated from the CST partitioning algorithm more cohesive. This will result in better recommendations as the hidden relationship between the users is also expressed by an edge. Now here we show a small example of how the derived trust heuristic can be used to generate new edges in the graph.

Let us consider the graph shown in Fig. 5, and let us say that if weight of edge between the users is greater than 0.75, then they have positive correlation.

Then by applying derived trust heuristic, we derive new edges as shown in Figs. 5a–c. Here derived edge's weight is simply taken as average of generator edges (edges that participate in forming derived edge). Hence, at first edge *AC* is derived from the edge *AB* and edge *CB*. Note that the edges *CD* and *DE* preexist in the graph but are not shown in Fig. 5 for clarity. Note that when we encounter edge *AD* and edge *DE*, no derived edge is generated as weight of $DE <= 0.75$. Hence, edge *AE* is not formed.

2.3 Online Heuristic

An *Online algorithm* is one that can process its input piece by piece in a serial fashion, i.e., in the order that the input is fed to the algorithm and output is generated without the knowledge of the entire input [5].

In Sect. 2.3.1, we describe the online heuristic that we propose to assign a partition to each streaming edge on the go. Then in Sect. 2.3.2, we discuss the need of using our online algorithm on top of CST partitioning algorithm and some use cases of our algorithm.

2.3.1 Updating the Clusters

The streaming edges are taken as input, and each edge will fall in one of the five cases. Let us say that the edge arrived has first node named as u and second edge as v.

Case 1. Both u and v are not present in any cluster. In this case, we make a new cluster consisting of that single edge. But if the total number of clusters exceeds the maximum number of clusters thresholds K (e.g., 256), then we just from that edge.

Case 2. When u belongs to some cluster C, v is not present in any cluster. In this case, add v to cluster C. If the cluster size of C exceeds the maximum cluster size threshold, then split the graph into 2 partitions, using, e.g., Kernighan Min Cut algorithm.

Case 3. When u is not present in any cluster, v is present in some cluster in C. The steps to be taken are similar to second case, by adding u to cluster C (where v is present).

Case 4. When u is present in some cluster C_1, v is present in some cluster C_2. In this case, we compare the size of the clusters C_1 and C_2. We merge the smaller cluster to the larger cluster. But if the size of both the clusters is comparable and the total size of both clusters exceeds the maximum size of the cluster size, then we drop the edge.

Case 5. When u is present in a cluster C, v is also present in the same cluster. In this case, we simply add the edge.

Now after discussing the above five cases, below we provide the algorithm for the online heuristic.

Online Heuristic (Called for each float in edge)

```
OnlineHeuristic (source, destination, weight)
If (A and B do not belong to any cluster)
    If (no_of_cluster < K )
       Make a new cluster C and add both nodes to C.
    Else
       Drop the Edge

If (A belongs to some cluster C1 and B does not belong to any cluster.)
    We add node B to C1.
    Then check
    if (size_of_C1 > threshold size)
        Apply graph_partition algorithm.

If (A does not belong to any cluster and B belongs to some cluster C1)
    We add node A to C1.
    Then check
    if (size_of_C1 > threshold size)
        Apply graph_partition algorithm.

If (A belongs to a cluster C1 and B belongs to some other cluster C2)
    Compare the sizes of cluster C1 and C2.
    If (one of clusters is smaller than the other and summed size of both cluster < TS)
       smaller cluster will join the bigger one.
    If (adding these clusters size of new cluster is more than the threshold_size(TS) )
        then we drop this edge

If (A belongs to a cluster C1 and B belongs to same cluster C1)
       Add the edge to C1.
```

2.3.2 Combining the Two: CST and Online Heuristic

Online Heuristic standalone. The problem with implementing the online heuristic standalone is we need a set of 'K' initial seed clusters so that the incoming edges can fall into the five cases with equal probability. But if the online heuristic is applied with empty clusters, then most of the floating-in edges fall in 'CASE 1,' i.e., making a new cluster, then we will have large number of clusters with single edges. Then after the total number of clusters exceeds the number of clusters threshold, then the further edges that fall in 'CASE 1' will be dropped, resulting in loss of many edges.

Online Heuristic with CST. To solve the above problem, we initially make the seed clusters with the existing nodes/users using CST and once we have formed 'K' equal size coherent clusters, we switch to online heuristic for real-time performance.

Note that the initial '*K*' seed clusters can be generated using any offline graph partitioning algorithm though we have used CST partitioning algorithm.

Outlined below is the overall algorithm that we propose for the online graph partitioning starting with the edge weight calculation, then applying the CST partitioning algorithm, then applying the derived trust heuristic followed by application of the online heuristic.

Overall Algorithm:
1: Edge Weight Calculation 2: Formation of CST using previous data 3: Make 'K' seed clusters from CST 4: Run Derived Trust Heuristic (DTH) on each cluster 5: Then switch to Online Heuristic for dynamically floating in edges [NOTE: if the edge that floats in is already derived by DTH, update weight by adding Friendship Bias Choose threshold size of clusters as per the time interval between resetting/switching servers. Assumption: all floating in edges will be distributed uniformly among 'K' clusters] 6: when cluster size is close to threshold, start preparing another server to switch to, repeat again from 1.

3 Experiments

3.1 Dataset

We have used the Foursquare dataset (1.5 GB) to experiment with our algorithms. Foursquare is basically an application which recommends hotels to their customers based on their friends and location. This dataset contains 2,153,471 users, 1,143,092 venues, 1,021,970 check-ins, 13,049,245 social connections, and 2,809,581 ratings that users assigned to venues.

3.2 Edge Weight Calculation

Now for edge weight calculation, ratings (for the similar rating information) and social graph (for their friendship information) tables were used.

In the dataset, rating history of some users is missing so when we try to calculate the edge weight, if any of the users do not have rating history, then neither correlation (*CoR*) nor common location ratio (*CLR*) can be calculated; hence, we simply take edge weight of those edges corresponding to the weight of zero correlation. These edges are invalid for edge weight calculation.

Table 1 Statistic showing number of positive, negative and neutral pairs/edges post edge weight calculation

Number of total edges, valid for edge weight calculation	6,316,205
Number of negative correlation user pairs/edges	99,004
Number of zero correlation user pairs/edges	5,953,765
Number of positive correlation user pairs/edges	263,436

In Table 1, we show some statistics about the classified correlation range and the corresponding number of edges, falling in those ranges:

After CST partitioning. During the experiment, we used approximately 50% of the social graph to form the CST and it can be shown that when seed clusters were being created from CST, edges representing negative correlation between pair of users were getting higher priority to get cut and form seed clusters. Also, these edges with negative correlation carry least importance when it comes to recommendation. Edges with positive correlation were rarely cut during cluster formation, saving the important connections between the users.

3.3 Derived Trust Heuristic and Online Heuristic

Derived Trust Heuristic is used to get connection between the users who are not friends but have high probability of having similar choice/taste (positive correlation). As per experiment, we found that many of the new generated connections/edges were having positive correlation and few of them were as good as those having best positive correlation among the original edges. When derived edge and original edge have the same correlation, friendship bias played an important role in breaking the tie.

Online Heuristic. After floating in edges from the rest approximately 50% of the social graph edges, it can be shown that the edges were uniformly distributed across the clusters.

4 Conclusion

In our work, we have taken care of the most essential factor that leads to best recommendation, that are persons with similar choice/taste.

(1) In Edge Weight Calculation formula, correlation is used to quantify this factor.
(2) Derived trust heuristic is used to get connection between the users who are not friends but have high probability of having similar choice/taste.

Our approach cleverly reduces the cluster sizes for recommendation computations, by partitioning the huge graph into small clusters on the basis of social network connections and similarity between the users. We also proposed a way to get connections between the users sharing similar choice but are not friends.

Our approach cleverly preserves the important connections between the users and forms small clusters without losing important information making the partitions more coherent to tackle the scalability issue in recommender system. Any recommender system can work on each cluster independently to generate recommendations.

References

1. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst. **43**(2), 618–644 (2007)
2. Jøsang, A., Gray, E., Kinateder, M.: Simplification and analysis of transitive trust networks. Web Intell. Agent Syst. **4**(2), 139–161 (2006)
3. Wang, G., Wu, J.: FlowTrust: trust inference with network flows. Front. Comput. Sci. China **5**(2), 181–194 (2011)
4. Filippidou, I., Kotidis, Y.: Online and on-demand partitioning of streaming graphs. In: IEEE International Conference on Big Data, Santa Clara, CA, pp. 4–13 (2015)
5. Albers, S.: Online algorithms: a survey. Math. Program. **97**(3), 3–26 (1997)
6. Jiang, W., Wang, G., Bhuiyan, Z. A. Md., Wu, J.: Understanding graph-based trust evaluation in online social networks: methodologies and challenges. ACM Comput. Surv. **49**(1) (2016)
7. Halder, S., Gupta,P.: Scalable spatially and socially aware collaborative filtering. In: International Conference on Emerging Trends in Computational and Applied Mathematics, India (2014)
8. Ye, M., Yin, P., Lee, W. C.: Location recommendation for location-based socia. In: Proceedings of the 18th SIGSPATIAL International (2010)
9. Bakshy, E., Rosenn, I., Marlow, C., Adamic, L.: The role of social networks in information diffusion. In: Proceedings of the 21st ACM International Conference on World Wide Web (WWW), pp. 519–528 (2012)
10. Caverleea, J., Liu, L., Web, S.: The social trust framework for trusted social information management: architecture and algorithms. Inf. Sci. **180**(1) (January 2010), 95–112 (2010)
11. Chen, S., Wang, G., Jia, W.: k-FuzzyTrust: an efficient trust computation for large-scale mobile social networks using fuzzy implicit social graph. Elsevier Inf. Sci. https://doi.org/10.1016/j.ins.2014.09.058 (2014)
12. Golbeck, J.: Computing and applying trust in web-based social networks. Ph.D. Dissertation, University of Maryland (2005)
13. Grandison, T., Sloman, M.: A survey of trust in internet applications. IEEE Commun. Surv. Tutor. **3**(4), 2–16 (2000). https://doi.org/10.1109/COMST.2000.5340804

Fuzzy Rough Set-Based Feature Selection with Improved Seed Population in PSO and IDS

Tarun Maini, Abhishek Kumar, Rakesh Kumar Misra and Devender Singh

Abstract In this paper, fuzzy lower approximation-based fuzzy rough set is used for selection of features. A distributed sampling (DS)-based initialization method is introduced to pick better seed population, in particle swarm optimization (PSO) and intelligent dynamic swarm (IDS). PSO and IDS are used for simultaneously selecting the appropriate feature subset. Fitness function for these computations is fuzzy rough dependency measure. Using the proposed initialization, while using PSO and IDS, improvement in size of selected subset of features with improved classification accuracy is also demonstrated.

Keywords Rough set · Fuzzy rough set · Intelligent dynamic swarm · Particle swarm optimization · Feature selection

1 Introduction

Feature selection process selects a subset of features which should be sufficient for the classification. When a large input data is suspected to have irrelevant and redundant features, then the input data needs reduction in terms of such features [1], since redundant features are of no use rather sometimes the irrelevant features may produce noise and increase the classification error in the data, and inclusion of the complete set may deteriorate the computational efficiency. It is expected that the feature reduction techniques will select such a feature subset which will contain

T. Maini · A. Kumar (✉) · R. K. Misra · D. Singh
IIT(BHU), Varaansi, Varanasi 221005, Uttar Pradesh, India
e-mail: abhishek.kumar.eee13@iitbhu.ac.in

T. Maini
e-mail: tmaini.rs.eee13@iitbhu.ac.in

R. K. Misra
e-mail: rkmishra.eee@iitbhu.ac.in

D. Singh
e-mail: dsingh.eee@iitbhu.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_11

the relevant information from the input data. Feature selection often improves classification accuracy by removing superfluous and irrelevant features. The selected feature set must be capable of predicting the decision feature in the same way as it would predict for a full dataset.

A number of techniques are proposed in literature that can be used to select feature [2–9]. But, not a single method can guarantee to produce an optimal feature set because the selection process depends on the problem at hand.

In this paper, feature selection is performed using fuzzy rough set [5], which is the combination of fuzzy sets and rough sets [10], discussed in the subsequent sections. The exhaustive search of appropriate subset of features is not feasible, because it involves lots of computational time and effort, which can be minimized by using other search techniques like evolutionary algorithms such as PSO and/or IDS. In this paper, we have implemented PSO [4] and IDS [6, 9], which facilitate simultaneous selection of feature subset. Since feature selection is essentially an optimization problem, we optimize the fuzzy rough dependency measure as used in fuzzy rough set. Further, traditionally random initialization of seed population is being used in PSO and IDS for feature selection. In this paper, an initialization method, which effectively picks the appropriate feature subset from a given feature set, has been proposed. We explored the possibility of a new initialization method for PSO and IDS. The proposed method of initialization improved the search process for effective feature selection.

2 Background

2.1 Rough Set

Feature selection can be done effectively using rough set theory (RST) [10, 11]. In case of conventional RST [10], a pair of sets formally approximates a given crisp set. This pair consists of lower and upper approximations. These two approximations are crisp or conventional sets.

Also in RST, facts hidden in data are analyzed and a minimal knowledge representation can be achieved. Besides the original data available in the given dataset, no other parameters of information such as expert knowledge or threshold values are required for using this technique.

Through examining equivalence relations produced by sets of features, the reduction of features is obtained. Features are reduced in such a manner that the chosen set produces the equivalent (or with minimal loss) predictive capability of the class label feature D as the unreduced feature set.

The reduct R is expressed as a reduced subset of the original feature set C. A dataset will contain several reduct sets; i.e., they may not be unique. A core is defined as the intersection of all reduct sets.

2.2 Fuzzy Rough Sets [5, 12]

Fuzzy rough sets allow greater flexibility in handling uncertainty. Here, elements may have membership between 0 and 1. Fuzzy rough set deals with both type of uncertainty, viz. indiscernibility (for rough sets) and vagueness (for fuzzy sets) [5, 12].

Fuzzy Lower approximation-based Fuzzy Rough Feature Selection (L-FRFS) A new feature selection technique named as fuzzy lower approximation-based fuzzy rough feature selection (L-FRFS) has been introduced in [5].

To determine fuzzy equivalence classes, L-FRFS uses fuzzy partitioning of the input space. Fuzzy lower and fuzzy upper approximations are defined as under:

$$\mu_{\overline{R_P}M}(m) = \sup_{n \in U} I\left(\mu_{R_P}(m,n), \mu_M(n)\right) \tag{1}$$

and,

$$\mu_{\underline{R_P}M}(m) = \inf_{n \in U} I\left(\mu_{R_P}(m,n), \mu_M(n)\right) \tag{2}$$

Here, fuzzy implicator I can be represented as $(min(1, 1 - m + n))$ and fuzzy similarity matrices (FSMs) denoted by R_p, which is calculated by Eq. (3) [12]

$$\mu_{R_P}(m,n) = T_{b \in P}\left\{\mu_{R_b}(m,n)\right\} \tag{3}$$

where T is a t-norm represented as $(max(m + n - 1, 0))$, $\mu_{R_b}(m,n)$ shows the degree to which objects m and n are similar for feature b.

FSM can be computed using the following equation.

$$\mu_{R_b}(m,n) = max(min(\frac{b(n) - b(m) + \sigma_b}{\sigma_b}, \frac{b(m) - b(n) + \sigma_b}{\sigma_b}), 0) \tag{4}$$

where σ is the standard deviation of attribute b.

Fuzzy positive region may be defined as:

$$\mu_{POS_{R_P}(Q)}(m) = \sup_{M \in U/Q} \mu_{\underline{R_P}M}(m) \tag{5}$$

The fuzzy rough dependency measure $\gamma'_P(Q)$ may be defined as follows:

$$\gamma'_P(Q) = \frac{\Sigma_{m \in U} \mu_{POS_{R_P}(Q)}(m)}{|U|} \tag{6}$$

A fuzzy rough reduct, R, preserves the degree of dependency of the unreduced dataset, i.e., $\gamma'_R(D) = \gamma'_C(D)$ where C denotes the conditional features set.

Further, for these computations discrete dataset is not necessary [13, 14].

The fuzzy connectives chosen throughout this paper and in this example are Lukasiewicz fuzzy implicator represented as $(min(1, 1 - m + n))$ and Lukasiewicz t-norm represented as $(max(0, m + n - 1))$.

In order to find the detailed description of these computations, reader may refer to [5].

3 L-FRFS Using PSO and IDS

In this work, PSO and IDS have been applied and discussed subsequently for simultaneous feature selection, and the fitness functions have been computed using fuzzy rough dependency measure of L-FRFS suggested in [5, 9, 15]. The method gives the string of selected features resulting in maximum fitness function.

The string of 0s and 1s is considered as a solution or particle. Index of each '1' shows that feature of corresponding index is selected, and index of each '0' means that the feature of the corresponding index is not selected. The string length equals the features existing in the unreduced dataset being used.

All the particles are initialized randomly, and fuzzy lower approximation for all classes is computed using Lukasiewicz fuzzy implicator. In the next step, the fuzzy positive regions are computed. Consequently, fuzzy rough dependency measure is computed and used as a part of the fitness function.

3.1 PSO-Based L-FRFS

To compare the effectiveness of the proposed method, feature selection has been performed using fuzzy rough dependency measure-based fitness function [5], with randomly initialized and proposed initialized methods of PSO-based search techniques.

The algorithm for PSO L-FRFS is discussed subsequently; only the fitness function will be fuzzy rough dependency measure as discussed in Eqs. (6) and (14) and Sect. 5.5.

Fundamentals of PSO PSO is based on swarm intelligence, inspired from the social behavior of fish schooling and bird flocking [16–18]. In PSO, we initialize probable solution, called population, randomly with several particles. Each particle tries to move toward the particles having global and local best values with some velocity, based on the optimization of fitness function.

$$V_i^{k+1} = w * V_i^k + r_1 * C_1(P_i^k - X_i^k) + r_2 * C_2(P_G^k - X_i^k) \tag{7}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{8}$$

where V is velocity, and w is inertia weight, C_1 and C_2 are constants, r_2 and r_1 are uniformly distributed random numbers. Normally we use $C_1 = C_2 = 2$, $pbest$ denotes the historical best for a particular particle itself, $gbest$ is the neighborhood best position, i.e., best of $pbests$, P_i denotes the ith particle, and X is the position of that particle. We have used the PSO algorithm suggested in [4].

Modified PSO for feature selection Wang et al. [4] suggested a modified PSO for feature selection. According to this work, velocity is obtained using the following equation.

$$\begin{aligned} V_i = & round(w * V_i + r_1 * C_1 * velocityupdate(Pbest_i - X_i) \\ & + r_2 * C_2 * velocityupdate(Gbest - X_i)) \end{aligned} \tag{9}$$

where V_i is an integer value. Function $velocityupdate(Gbest - X_i)$ returns an integer.

The value of V_i obtained in this function is defined as follows

$$V_i = \begin{cases} 1; \text{ if} V_i \leq 1 \\ round(N/3); \text{ if} V_i > N/3 \\ unchanged; \text{ otherwise} \end{cases} \tag{10}$$

where N is the total number of features. Using the above value of V_i, particle X_i is updated using the equation as shown below

$$X_i = positionupdate(Gbest, X_i, V_i) \tag{11}$$

For the detailed working mechanism of functions *velocityupdate* and *positionupdate*, reader can refer to [4].

Algorithm Algorithm for PSO used for feature selection is as follows:

(1) Initialize the swarm, collection of various particles (solutions) randomly and parameters as discussed in Sect. 5.4.
(2) Compute the fitness function using Eq. (14) for each solution.
(3) For every solution, compare its pbest value with its current fitness value. If the fitness value of pbest is lower than the fitness value of the current value, then current particle's position x_i replaces the P_i and pbest's fitness value is replaced by current particle's fitness value.
(4) Define the particle, having the maximum fitness value till now, i.e., best of pbest values, as global best value, gbest and, its position as P_g.
(5) For all the particles, updatation of the positions and velocities are done by using appropriate equations discussed in Sect. 3.2.
(6) Repeat steps 2–5 for 100 generations.

3.2 IDS-Based L-FRFS

Intelligent dynamic swarm (IDS) [6, 9] has been suggested for simultaneous feature selection along with the fitness function computed using dependency measure defined in L-FRFS [5].

Fitness function is fuzzy rough dependency measure as discussed in Eqs. (6), (14) and Sect. 5.5.

Fundamentals of IDS This method is similar to PSO, and it proceeds without velocity term as defined in PSO. In this method again we initialize the population (probable solutions) randomly. The following parameters are required: $C_w = 0.1$, $C_p = 0.4$, $C_g = 0.9$, and random number r is generated.

Algorithm Algorithm for IDS is as follows:

(1) Initialize the swarm randomly and take parameters as per Sect. 5.4.
(2) Compute the fitness function using Eqs. (6) and (14).
(3) For every particle (solution), compare its pbest value with its current fitness value, if it is greater than pbest value then set or replace current fitness value as pbest.
(4) Define the particle, having the maximum fitness value till now, i.e., best of pbest values, as global best value, gbest and its position as P_g.
(5) For each particle, between 0 and 1, a random number r is generated. If the value of r is in between 0 and C_w, then the current particle will not be updated. Otherwise if r is smaller than the C_p but greater than the C_w, then the current particle will be updated by the particle having pbest value. Otherwise if $C_p \leq r < C_g$, then the current particle will be replaced by the particle having gbest value. Otherwise if $C_g \leq r < 1$, then the current particle will be replaced by the particle generated randomly.
(6) Repeat steps 2–5 for 100 generations.

4 Proposed Method

For extending the capability shown by PSO [4] and IDS [6, 9], new methods for feature selection have been proposed.

In this work, PSO and IDS with a new initialization technique have been implemented, which gives improved performance. This initialization attempts to explore the search space in a distributed fashion.

This initialization technique distributes the search space in three parts and then takes solutions (samples) randomly from each of the three search spaces. Hence, this technique is referred to as distributed sample initialization (DS initialization) in this paper, to differentiate it from the random initialization (RANDOM initialization) used conventionally.

4.1 DS Initialization

A new method of random initialization is proposed by which if the true reduct of dataset contains less, medium, or large number of features, then it will pick the suitable number of features early from our initialization method, which will prevent from getting stuck to the problem of local minima and will provide particles very close to their final solution in early generations.

In the proposed initialization method, first one-third population is initialized as biased toward picking more number of features, second one-third population is initialized as biased toward picking medium number of features, and the last one-third population is initialized as biased toward picking less number of features.

Thus, the probability of the presence of near-optimal solution in the population would be higher, and due to this, iteration-by-iteration convergence of population toward optimal solution would be faster, preventing problem of getting stuck to local minima.

The proposed method of initialization provides the distributed samples of particles with better variation in selected number of features, i.e., too low, medium, too high.

4.2 PSO and IDS with DS Initialization

In the present work, PSO [4] and IDS [6, 9] have been implemented with the DS initialization, as discussed in Sect. 4.1 in place of random initialization; rest all other assumption and parameter remain the same. Results have been compared with that of random initialization RAND initialization. The performance of PSO and IDS with proposed DS initialization is better.

5 Experimental Setup

We implemented all the feature selection methods used in this paper in MATLAB version 8.1 (R2013a). The computer is Intel core i3 processor 1.7 GHz, with 2 GB RAM, and the system is Windows 8.1. Assumptions considered and parameters related to experiment and data are discussed as follows:

5.1 Dataset

All the benchmark datasets are taken from the UCI data repository of machine learning [19]. Characteristics of datasets are described in Table 1.

Table 1 Characteristics of datasets

S. No.	Dataset	No. of objects	No. of features	Unreduced classification accuracy	
				Classifier: J48	Classifier: JRip
1	Ecoli	336	7	84.23	80.36
2	Ionosphere	351	34	91.45	88.6
3	Wine	178	13	93.82	91.57
4	LSVT	126	310	76.98	78.57

5.2 Data Normalization

In the present work, any dataset is normalized between 0 to 1, both inclusive; using the following equation, the data are scaled into range [0, 1].

$$L_i^{norm} = \frac{L_i^t - L_i^{min}}{L_i^{max} - L_i^{min}} \tag{12}$$

where L_i^{norm} is normalized value of ith element of a given feature, and L_i^{min} and L_i^{max} are, respectively, minimum and maximum values of all the elements of the feature.

5.3 Data Preprocessing

Fuzzy similarity matrices (FSMs) have been computed using Eq. (4) for all the features of the corresponding datasets individually. These FSMs are used to compute fuzzy rough dependency measure corresponding to feature set of more than one feature.

5.4 Parameter Setting

In this section, parameters used for PSO and IDS are described. The values of these parameters are usually taken on the basis of trial and error.

In PSO [4] Eq. (7), C_1 and C_2 will have their usual value 2, and w the inertia weight will decrease from 1.4 to 0.4 generation by generation.

$$w = (w - 0.4) * \left(\frac{Max_Iter - Current_Iter}{Max_Iter} \right) + 0.4 \tag{13}$$

Parameters in IDS [6] are set as follows: $C_w = 0.1$, $C_p = 0.4$, $C_g = 0.9$.

5.5 Fitness Value

The fitness value will depend upon $\gamma'_P(Q)$ as shown in Eq. (6). Final fitness value is computed using the following formulae [4].

$$Fitness_Value = 0.9 * \gamma'_P(Q) + 0.1 * \left(\frac{|C| - |R|}{|C|}\right) \tag{14}$$

where $\gamma'_P(Q)$ is the fuzzy rough dependency measure of selected feature set R relative to decision feature D. C is the number of features in the unreduced dataset.

The fitness value is the weighted objective function of the two terms. This is done so that if two feature subsets R_1 and R_2 have same $\gamma'_P(Q)$, then one with R_1 takes the higher value when R_1 has the lower number of features than R_2.

5.6 Classification Accuracy

In the present work, classifiers J48 and JRip are used on each of the datasets to calculate S-10-FCV accuracy for comparison purposes. J48 is the Java version of the decision tree-based classification method C4.5 [20] algorithm, and JRip is based on learning propositional rules.

We computed the classification accuracy of J48 [21] and JRip [22] using data mining workbench WEKA [23, 24].

6 Results and Discussions

To compare the performance of feature selection, based on L-FRFS using PSO and IDS, with random initialization (reported in literature) and proposed method of initialization (DS initialization), both have been implemented.

Results of both of these initialization techniques on standard benchmark datasets have been presented in Tables 2 and 3. PSO and IDS using random and DS initialization have been executed 12 times each, for each of the datasets. Each run is of 100 generations, and population size is 100 for all the datasets. Mean and standard deviations (s.ds) of a number of selected features are calculated for these 12 runs for each of the cases.

In Tables 2 and 3, for *Ecoli* dataset, it is observed that there is no impact of DS initialization in the performance of PSO and/or IDS, because of the same results in

Table 2 Comparison of selected number of features for random and DS initialization

Dataset	Feature selection method	No. of features selected		
		Min	Mean	s.d.
Ecoli(7)	PSO-RANDOM	5	5	0
	IDS-RANDOM	5	5	0
	PSO-DS	5	5	0
	IDS-DS	5	5	0
Ionosphere(34)	PSO-RANDOM	6	7.33	0.88
	IDS-RANDOM	7	8.16	0.71
	PSO-DS	6	7	0.6
	IDS-DS	7	7.83	0.57
Wine(13)	PSO-RANDOM	4	4	0
	IDS-RANDOM	4	4	0
	PSO-DS	4	4	0
	IDS-DS	4	4	0
LSVT(310)	PSO-RANDOM	113	120.91	4.33
	IDS-RANDOM	119	122.59	1.92
	PSO-DS	13	16.5	2.11
	IDS-DS	16	20.75	2.86

every run, which is evident from zero s.ds. Classifier JRip improves the performance in terms of classification accuracy in each method.

In case of *Ionosphere* dataset (Tables 2 and 3), mean subset size further reduces with slightly improved classification accuracy, when using DS initialization in PSO and/or IDS. Classifier J48 facilitates the comparable classification accuracy while keeping it unreduced, whereas classifier JRip improves the classification accuracy in each method.

In the case of *Wine* dataset, all methods pick 4 features every time (Table 2), but the combination of features are different. DS initialized PSO is able to manage a more appropriate subset than randomly initialized PSO, as its mean classification accuracy is higher as shown in Table 3. Classifier J48 facilitates the comparable classification accuracy while keeping it unreduced, whereas classifier JRip improves the performance in terms of classification accuracy in each method.

From Table 2, it is also observed that there are major improvements in case of the *LSVT* dataset where the number of features in unreduced dataset is 310. Traditional random initialization reduces minimum number of features selected up to a minimum of 113 for PSO. But using DS initialization, the minimum number of features selected gets reduced up to 13 for PSO techniques, while the accuracy remains comparable. Traditional random initialization reduces minimum number of features selected up to a minimum of 119 for IDS; whereas using DS initialization, the minimum number

Table 3 Comparison of classification accuracy for random and DS initialization using classifiers J48 and JRip

Dataset	Feature selection method	Classification accuracy using J48				Classification accuracy using JRip			
		Unreduced	Max	Mean	s.d.	Unreduced	Max	Mean	s.d.
Ecoli(7)	PSO-RANDOM	84.23	82.44	82.44	0	80.36	81.25	81.25	0
	IDS-RANDOM		82.44	82.44	0		81.25	81.25	0
	PSO-DS		82.44	82.44	0		81.25	81.25	0
	IDS-DS		82.44	82.44	0		81.25	81.25	0
Ionosphere(34)	PSO-RANDOM	91.45	91.74	89.52	1.35	88.6	91.45	89.64	1.25
	IDS-RANDOM		93.73	90.1	2.04		92.02	89.19	1.86
	PSO-DS		91.74	89.58	1.73		91.74	89.07	1.79
	IDS-DS		93.73	89.86	2.15		92.02	89.31	1.82
Wine(13)	PSO-RANDOM	93.82	93.82	91.56	2.94	91.57	92.13	90.4	1.71
	IDS-RANDOM		93.82	93.16	0.67		92.13	90.63	1.04
	PSO-DS		93.82	93.07	0.8		92.13	90.91	1.08
	IDS-DS		93.82	93.54	0.5		91.57	90.31	0.75
LSVT(310)	PSO-RANDOM	76.98	80.16	75.13	3.4	78.57	82.54	77.18	2.69
	IDS-RANDOM		79.37	74.94	3.11		86.51	78.7	2.96
	PSO-DS		82.54	73.14	5.12		80.95	74.2	3.93
	IDS-DS		80.95	75.52	4.19		83.33	75.13	3.84

of features selected gets reduced up to 16 for IDS techniques, while the accuracy remains comparable.

7 Conclusion

On the basis of above results (Tables 2 and 3) and discussions, it can be concluded that PSO and IDS technique with DS initialization give improved performance with minimum features obtained and classification accuracy achieved. For all the datasets, PSO and IDS with DS initialization are able to maintain or increase the unreduced classification accuracy, but PSO and IDS with random initialization sometimes fail to manage the same.

It is also observed that DS initialization has yielded drastic reduction in the minimum and mean number of features selected when original dataset has big number of features (as observed in *LSVT*). Thus, it is concluded that DS initialization significantly improves performance of PSO and IDS techniques when the number of features is large.

References

1. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
2. Ren, D., Ma, A.Y.: Research on feature extraction from remote sensing image. In: International Conference in Computer Application and System Modeling, vol. 1, pp. V144–V148 (2010)
3. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007)
4. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature selection based on rough sets and particle swarm optimization. Pattern Recognit. Lett. **28**(4), 459–471 (2007)
5. Jensen, R., Shen, Q.: New approaches to fuzzy-rough feature selection. IEEE Trans. Fuzzy Syst. **17**(4), 824–838 (2009)
6. Bae, C., Yeh, W.-C., Chung, Y.Y., Liu, S.-L.: Feature selection with intelligent dynamic swarm and rough set. Expert Syst. Appl. **37**(10), 7026–7032 (2010)
7. Verma, N.K., Maini, T., Salour, A.: Acoustic signature based intelligent health monitoring of air compressors with selected features. In: 2012 Ninth International Conference on Information Technology: New Generations (ITNG), pp. 839–845. IEEE (2012)
8. Maini, T., Misra, R.K., Singh, D.: Optimal feature selection using elitist genetic algorithm. In: IEEE Workshop on Computational Intelligence: Theories, p. 2015. Applications and Future Directions (WCI), IEEE (2015)
9. Maini, T., Kumar, A., Misra, R.K., Singh, D.: Feature selection with intelligent dynamic swarm and fuzzy rough set. In: IEEE International Conference on Computing, Communication and Automation 2017 (ICCCA 2017)
10. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**(5), 341–356 (1982)
11. Swiniarski, R.W., Skowron, A.: Rough set methods in feature selection and recognition. Pattern Recognit. Lett. **24**(6), 833–849 (2003)
12. Mac ParthaláIn, N., Jensen, R.: Unsupervised fuzzy-rough set-based dimensionality reduction. Inf. Sci. **229**, 106–121 (2013)

13. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets*. Int. J. Gen. Syst. **17**(2–3), 191–209 (1990)
14. Wang, C., Qi, Y., Shao, M., Hu, Q., Chen, D., Qian, Y., Lin, Y.: A fitting model for feature selection with fuzzy rough sets. IEEE Trans. Fuzzy Syst. (2016)
15. Mac Parthaláin, N., Jensen, R.: Fuzzy-rough feature selection using flock of starlings optimisation. In: 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–8. IEEE (2015)
16. Eberhart, R.C., Kennedy, J. et al.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science
17. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: The 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, pp. 69–73. IEEE (1998)
18. Shi, Y. et al.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 81–86. IEEE (2001)
19. Lichman, M.: UCI machine learning repository (2013) [Online]. http://archive.ics.uci.edu/ml
20. Quinlan, J.R.: C4.5: Programming for Machine Learning. Morgan Kauffmann (1993)
21. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
22. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123 (1995)
23. Witten, I., Frank, E.: Data mining: practical machine learning tools with java implementations. In: Kaufmann, M. (ed.) San Francisco (2000)
24. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newslett. **11**(1), 10–18 (2009)
25. Kohavi, R., et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection. IJCAI **14**(2), 1137–1145 (1995)

Domain-Independent Sentiment Analysis in Malayalam

V. Kasthoori, B. Soniya and V. Jayan

Abstract This paper is focused on the research in domain-independent sentiment analysis in Malayalam language. A sentence-level sentiment analysis using machine learning method and fuzzy logic is used for the work. Using ART classifier, the dataset is trained and monograms in the dataset are divided into eight classes and assigned a score. When the system accepts the input text, monograms in the text are used to find the domain of the text. After identifying the domain by comparing the trained data, the system assigns the scores to the sentences based on the scores of the pre-trained data. The overall score is calculated by adding up the individual scores and identifies whether the text is having positive or negative sentiments. By using fuzzy logic, the polarity of the input is calculated. Since there is no standardized dataset in Malayalam for sentiment analysis, the training data is collected from the Malayalam online newspaper. In short, system identifies the domain of the input text and then finds the sentiment and its polarity.

Keywords Sentiment analysis · Fuzzy logic · Association rule tree

1 Introduction

The Internet has been a continual source of news, entertainment, and education for users around the world for more than 20 years. As the Internet grows, humans depend on it for gathering opinions and information on various products and services based

V. Kasthoori (✉) · B. Soniya
Sree Chitra Thirunal College of Engineering, Trivandrum, India
e-mail: kasthooriv@gmail.com

B. Soniya
e-mail: soniya@sctce.ac.in

V. Jayan
Centre for Development of Advanced Computing, Trivandrum, India
e-mail: jayan@cdac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_12

on which decision is made. The attitude of an individual who shares opinions about any specific area can be found out by using sentiment analysis [1].

Sentiment analysis also called as opinion mining uses text analysis, computational linguistics, and natural language processing methods for determining the feelings from the text. Generally, the text is categorized into positive, negative, or neutral. Natural languages are normally used for expressing sentiments, and electronic device like computers used for processing is a complex task. Some of the challenges of sentiment analysis in general are implicit sentiment and sarcasm, domain dependency, thwarted expectations, subjectivity detection, pragmatics, world knowledge, entity identification, negation, etc. [2].

Many works exist in sentimental analysis of English language text. Due to characteristics of English language like simplicity of inflection, fixed word order, use of periphrases, development of intonation, sentiment analysis in English is comparatively easier than any language.

Malayalam is a South Indian language which is nominated as one of the classical languages in India in 2013. The Dravidian languages (Malayalam, Tamil, Telugu, Kannada) are considered as the most agglutinative languages. It is an inflectionally rich language with a free word order [3]. In free word order languages, the order of the words carries only secondary information, while the gross meaning is contained elsewhere.

Sentiment analysis in Malayalam is a more challenging task. Some of them are (i) most of the user comments or reviews in Malayalam on the Internet are in transliterated form in English and (ii) user comments or reviews are a combination of English and Malayalam. In these situations, it becomes difficult to identify the language and hence sentiment analysis in Malayalam is considered as tough task. Another challenge faced in research on sentiment analysis in Malayalam is the lack of a standard dataset for training purpose.

This work proposed a domain-independent sentiment analysis in Malayalam. In this work, we focused on sentence-level sentiment analysis. Machine learning and fuzzy logic methods are used for sentiment analysis. Using ART classifier, the dataset is trained and monograms in the dataset are divided into eight classes and assigned a score. When the system accepts the input text, monograms in the text are used to find the domain by comparing the trained data and scores are assigned. The individual scores are added up to find the total score, and fuzzy logic is used to find the polarity of the input. A corpus consisting of 400,000 words are manually created and trained.

This paper is organized into five sections as follows: Related works are described in Sect. 2. Section 3 covers the proposed methodology and the working of rules, and Sect. 4 portrays the results and discussion. Finally, Sect. 5 concludes this paper.

2 Literature Survey

Most of the research publications in English for sentiment analysis use machine learning-based methods, semantic orientation methods, rule-based methods, or fuzzy logic. The works on agglutinative languages like Malayalam are very less. Moreover, the existing works in Malayalam language are domain-specific.

Bhattacharyya's and Agarwal's [4] work presents a machine learning technique for the sentiment analysis in English for movie reviews. It uses Wordnet-based method on documents for the effective incorporation of linguistic information. Barbosa and Feng [5] propose a classification method specifically for Twitter which consists of two steps. They use noisy labels as training data. Firstly, the tweet was classified into subjective and objective, and then, they classified these subjective tweets as positive or negative.

Mohandas et al. [6] proposed a method to extract the mood from the Malayalam sentences using pointwise mutual information and information retrieval (PMI-IR). In their work, the part-of-speech tagging for each sentence collected from Malayalam novels is done and then semantic-oriented words such as adjective, adverbs are extracted. A reference word set for each emotion is created and used in this method. The semantic orientation mutual information and information retrieval (SOPMI-IR) formulas are modified in such a way that it could classify the input into four classes. Thus, using the modified SOPMI-IR formula, the extracted words, and reference word set, the semantic orientation of the input as well as the strength of the semantic orientation is found. It was concluded that the proposed method gives an accuracy of 63% only.

Nair's [7] work discusses a rule-based approach for sentiment analysis from Malayalam movie reviews. Due to inflectional properties of Malayalam, the author used Sandhi (joining) splitter before assigning positive or negative values for the words. After the tokenization process, the words are checked with the pre-stored words and assign sentiment for each word. Negation rules are used for extracting the sentiments of the input. It also handles smiley which can be used for expressing different emotions. An accuracy of 85% was achieved by this new approach.

Another work of Nair [8] discusses the sentiment analysis of Malayalam film review using machine learning technique. Support vector machine (SVM) and conditional random field (CRF) combined with rules are used for finding the sentiment analysis. This work finds the polarity of the review with the rating. A tag set consisting of 10 tags were built, and the corpus was manually tagged. The corpus was trained using SVM and CRF, separate models were built, and negation rules are used for finding the polarity and rating. This paper concludes SVM as more accurate than CRF.

Anagha et al. [1] proposed a hybrid approach for domain-specific sentence-level sentiment analysis. Training and tokenizing the data, tagging the tokens, applying special rules, and extracting mood using fuzzy logic are the different phases of this work. In the training phase, the dataset was divided into seven classes and score was assigned to each word manually. The inputs are tokenized in the tokenizing phase,

and in tagging phase, the appropriate tags are assigned to the input words. In certain cases, some special rules are applied and the individual score is summed up to obtain the overall score. Thus, obtained overall score is given to the triangular membership function and finds the amount of positivity and negativity. This work agrees a 91.6% precision rate, but it is dependent on a domain.

Thus, it is clearly understood that only few works have been done in Malayalam and all the existing works are domain-specific also. So in this work, a method is proposed for performing sentiment analysis which is independent of the domain.

3 Methodology

This work proposes a domain-independent sentiment analysis method in Malayalam using machine learning and fuzzy logic. An association rule tree (ART) classifier is used to identify the domain of input text, and by using fuzzy logic, the polarity of the input is calculated (positive, negative, or neutral). The method consists of training and testing phases.

3.1 Training Phase

The first step involved is to collect news and reviews from different domains and store as corpus. Mainly, the news and reviews are collected from different sites like www.mathrubhumi.com, www.manoramaonline.com. Next, the monograms (each word) in the document are extracted and they are stored and separated by their domains. An association rule tree (ART) classifier is used to train the data.

ART is a classifier which is used on huge datasets for classifying very accurately [9]. Association rule discovery methods are utilized by the ART classifier for building decision list and thereby ensuring a good scalability [9]. Instead of discovering rules one at a time, ART identifies multiple rules simultaneously, and using these sets of association rules, it builds partial classification models. The uncovered instances (not covered by the selected association rules) in the input dataset are grouped together in 'else' branches. Since partial classification models are built using association rule mining algorithms, it requires MinSup (minimum support for the frequent item sets) threshold, which is a percentage of the current dataset size, and it implicitly restricts the tree branching factor.

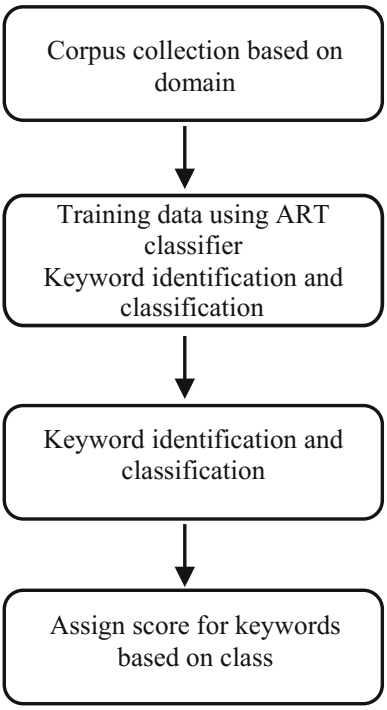
Then, the monograms in the given text are considered as the keywords and these are classified into eight different classes. These classes are depicted in Table 1.

Each keyword is assigned a score ranging from −1 to 1. The score 0 is assigned to the neutral words, while the score greater than 0 is assigned for positive words and less than 0 for negative words. The assigning score for each keyword in training phase is done manually. Thus for each domain, separate corpus is maintained. After

Table 1 Eight different classes

Class	Example	Description
Positive	'nallath'	Happy mood
Negative	'cheetha'	Sad mood
Neutral	'vartha'	Don't convey any mood
Inverse negative	'alla,' 'illa'	Inverse the mood of non-neutral word
Intensifier	'valare'	Intensifies the non-neutral word
Dilator	'kurach'	Dilates the non-neutral word
Special	'maduthu'	Gives negative sense if the preceding word is neutral
Connector	'ennalum'	Words used to connect two sentences

Fig. 1 Training phase



the training phase, the system takes the inputs. The steps involved in training phase are depicted in Fig. 1.

Table 2 Rules

Class	Rules
Inverse negative	Finds out the first word from any class (except neutral class) and changes the polarity of the word by changing the sign of the score
Intensifier	Finds the first non-neutral word after the word from Intensifier class and doubles the score of first non-neutral word
Dilator	Finds the first non-neutral word after the word from Dilator class and halved the score of the first non-neutral word is halved
Special	If the first-seen word before the word from Special class is neutral, then the sign of the score of special word is altered. The score of the special word is unchanged, if the first-seen word is non-neutral
Connector	The score of non-neutral words, after the words from Connector class is used to find the sentiments

3.2 Testing Phase

Testing is done with comments and reviews from multiple domains. The monograms from the input text are extracted, and using the ART classifier, the appropriate domain of the input is identified. After identifying the domain, sentiment analysis is performed. For finding the sentiments of the input, the predefined trained corpus is compared with the input and assigns scores for the input text. The total scores of sentences are calculated based on rules. Rules alter the score of words depending on the words preceding or following it. Rules are depicted in Table 2.

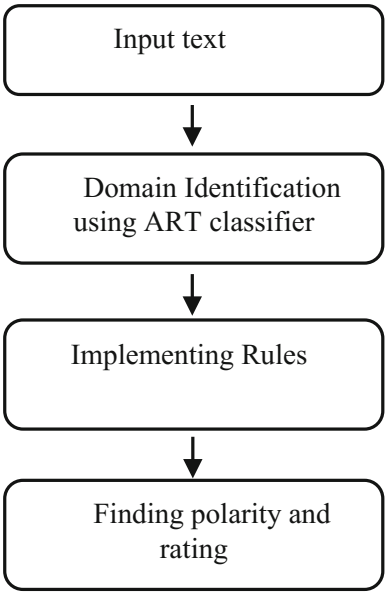
After rules are implemented, the total score of the input is calculated by adding up the scores of each word. As next step, the total score is given to triangular membership function. Triangular membership function is defined in Eq. 1.

$$f(a, x, , y, z) = max(min\frac{(a-x)}{(y-x)}, \frac{(z-a)}{(z-y)}, 0) \tag{1}$$

'x' and 'z' are the border values (−1 and 1), 'y' = (x + z)/2, and 'a' is the total score of the input.

The triangular membership function's output ranges from 0 to 1. A positive value for total score implies the input sentence to be positive, and a negative value for total score implies the sentence to be negative. The percentage of positivity and negativity in the input is calculated by using the fuzzy membership value of total score. Since the output of the triangular membership function ranges from 0 to 1, the value implies the mood to be neutral to highly saturate. This can be used for rating the review or news. The testing phase is depicted in Fig. 2.

Fig. 2 Testing phase



Example: Input: കോഴിക്കോട്ടെ വളരെ നല്ല ഒരു സ്കൂൾ ആണ് പ്രൊവിഡൻസ്. This input identifies domain as Education using the ART classifier. For sentiment analysis, the predefined trained corpus (education domain) is used to find the score of the monograms in the input text. As a result, the words വളെര and നല്ല are identified as intensifier and positive, respectively. The total score of the input is calculated after applying the rules, and it is given to triangular membership function.

4 Result and Discussion

4.1 Dataset Used

Since no standard datasets are available, to validate this work data collected from different online Web sites is used. Data is collected from the online Web sites like mathrubhumi.com, manorama.com. A large training corpus of 400,000 words is used for training the data. The data is collected domainwise, and the domains are Movie, Health, Food, Money, etc., and stored separately. The sample dataset is depicted in Fig. 3.

Fig. 3 Dataset

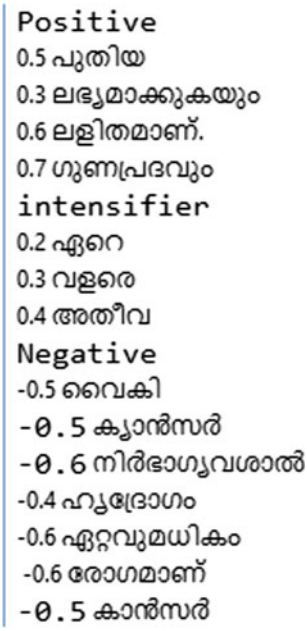


Table 3 Quantitative evaluation for domain identification

Precision	Recall	F_measure
87.8%	90.1%	88.9%

Table 4 Quantitative evaluation for sentiment analysis

Precision	Recall	F_measure
85.7%	92.3%	88.7%

4.2 Evaluation

The evaluation method consists of dividing the dataset into two: training and testing; 80% of collected data is used for training, while 20% is used for testing. During the training phase, the collected words are classified into eight classes and each word is manually scored between −1 and 1 as described in Sect. 3. The testing phase consists of domain identification and sentiment analysis. Precision, recall, and F_measure are the performance measures used to evaluate the performance of domain identification and sentiment classification. Quantitative evaluation for domain identification and sentiment analysis is depicted in Table 3 and Table 4, respectively.

i. Precision: Precision is percentage of all selected reviews that are correctly classified in class out of available reviews and is calculated using Eq. 2.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{2}$$

ii. Recall: Recall is percentage of correct reviews that are selected in class from all the available reviews and is calculated using Eq. 3.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{3}$$

iii. F_Measure: F_measure is harmonic mean of precision and recall and is calculated using Eq. 4.

$$\text{F_Measure} = 2\text{PR}/(\text{P} + \text{R}) \tag{4}$$

The precision and recall of the system are low compared to the works done on domain-specific [1]. The reason is that certain words have different meanings depending on the context. In certain situation, it becomes difficult to identify the correct context. Besides, same words may come in different domains. In such situations, the classification may go wrong. For example: (i) അർബുദം തടയാൻ കീഴാർനെല്ലി Normally, this sentence comes under health domain, but in certain situation, the system may sometimes identify the domain as agriculture which is not correct due to word കീഴാർനെല്ലി. (ii) രാമു അർജുനോട് ചെസ്സിൽ തോറ്റു The domain of this sentence may be identified as sports, and it is identified as negative sentence. But from the perspective of the second person (അർജുൻ), it is positive.

5 Conclusion and Future Works

Currently, all the existing works in Malayalam are domain-specific. Most of them are related to film review ratings. In this paper, a domain-independent sentiment analysis in Malayalam is carried out. This work also helps in identifying the domain of the input and the polarity of the input as well. It uses machine learning-based method, namely ART, to identify the domain, and fuzzy logic is used to find the polarity of the text. From the experimental results, it is clear that the proposed methodology is a very effective method for decision making. Though the results presented in this method are satisfactory, scope for improvement is still there. Future work can handle the challenges like Malayalam transliterated to English and the Malayalam reviews written in combination of English and Malayalam.

References

1. Anagha, M. et al.: Fuzzy logic based hybrid approach for sentiment analysis of Malayalam movie reviews. In: IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015. IEEE (2015)
2. Mukherjee, S., Bhattacharya, P.: Sentiment Analysis: A Literature Survey. Indian Institute of Technology, Bombay Department of Computer Science and Engineering (2012)
3. Idicula, S.M., David, P.S.: A morphological processor for malayalam language. South Asia Res. **27**(2), 173–186 (2007)
4. Bahattacharyya, P., Agarwal, A.: Sentiment analysis: a new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. In: Proceedings of the International Conference on Natural Language Processing (ICON), October 2005
5. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics (2010)
6. Mohandas, N., Nair, J.P.S., Govindan, V.: domain specific sentence level mood extraction from malayalam. In: 2012 International Conference on Text. Advances in Computing and Communications (ICACC). IEEE
7. Nair, D.S., Jayan, J.P., Sherly, E.: SentiMa-sentiment extraction for Malayalam. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE 2014
8. Nair, D.S., Jayan, J.P., Sherly, E.: Sentiment analysis of malayalam film review using machine learning techniques. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE (2015)
9. Berzal, F., Cubero, J.C., Sánchez, D., Serrano, J.M.: Art: a hybrid classification model. Mach. Learn. **54**(1), 67–92 (2004)

Path Planning for Multiple Mobile Robots by Priority Assignment

Ram Kishan Dewangan, Anupam Shukla and W. Wilfred Godfrey

Abstract Priority assignment in the multi-robot path planning becomes a key challenge in the present situation. The motivation for the priority assignment is to utilize the search space for all the robots and minimize the overall path length. When the number of robots increases, the complexity also increases for the same search space. This paper gives the idea about how to assign the priority and choose the best minimum path length. The simulation results show the output performed and success rate of the robots.

Keywords Multi-robot path planning · Static environment · Priority assignment

1 Introduction

One of the issues in the field of multi-robot path planning is priority assignment. In prioritized multi-robot path planning, each robot is given a priority according to which they perform their operation. The priority assignment is done so as to maximize the number of successful path completions and minimize the total path length. Suppose there is a narrow road and some robots are passing from opposite side to achieve their respective target. In this situation, there may be a collision between robots if no priority has given to robots. If we set the priority to the robots, then higher-priority robot will move first and proceed toward its goal; after that, lower-priority robots move.

R. K. Dewangan (✉) · A. Shukla · W. Wilfred Godfrey
Soft Computing and Expert System Laboratory, ABV – Indian Institute of Information Technology and Management, Gwalior, Gwalior, India
e-mail: ramadew7@gmail.com

A. Shukla
e-mail: anupamshukla@iiitm.ac.in

W. Wilfred Godfrey
e-mail: godfrey@iiitm.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_13

This is one way to give the solution, and another solution is to interchange the priority of the robots, so the second robot moves first and reaches to its destination followed by the first robot.

In this way, the priority for multiple robots can find the best solution for the specific set of sequences and displays the length of the minimum path.

2 State-of-the-Art Algorithm

In [1, 2], robots begin with an initially assigned priority strategy and the priority of the robots interchanged randomly for further path planning. If the new plan comes about with the shorter path length, then proceed with this plan. In this paper, researchers suggested randomized search procedure with hill-climbing method since basic hill-climbing method much of the time stalls out in local minima.

In [3, 4], the robots formed in two sets as S_1 and S_2. S_1 includes robots that do not contain any cycle. S_2 contains the combination of robots with a cycle, so here, to achieve the solution; a robotic request requires a random restart.

In [5, 6], collision-free and kinematic requirements of the robots are presented. A priority scheme is utilized in cooperative path planning described in the paper. The disadvantage in this approach is that higher-priority robots ignore the lower-priority robots. But in practical situation, one cannot ignore the same rather treat is as the obstacle.

In [7], the goal is to identify paths for various robots in a shared map from starting position to target position without clashing with different robots and static obstacles. The disadvantage in this paper is the approach discussed is iterative so cannot achieve parallelism. Another limitation is that it computes the path for those robots which gives a complete result only.

In [8], authors outlined an approach for multi-vehicle path planning. In this paper, needs of the operators can be depicted by a stage. Two methodologies are examined for path planning: Initially, the priority is picked randomly. Second, the general outcome is optimized. To solve the collision among the robots, tree search technique is proposed.

In [9], strategy was utilized as decentralized multi-agent motion planning technique. A* path planning calculation is utilized for detecting the path for the individual robots. This paper discusses path planning in 3D environments.

In [10], authors manage two sorts of priority framework—static and dynamic. In static priority framework, each robot is allocated priority before locating the path for the particular destination. The priority task happens during runtime and does the important swapping of the priorities of the robots as needed in dynamic priority framework. The authors take care of the issue in its own particular environment. Limitation of this approach is that it does not calculate priority based on global search space.

In [11], the author has suggested path planning to an objective robot development in a cluttered domain which contains unidentified obstacles. Authors created

dynamic priority methodology for different portable robots in speaking to the objective development. In this, enhanced RRT-based path organizer was produced for different portable robots as the calculation for path planning. The dynamic priority framework is proposed in this paper as in a few circumstances static priority does not work properly.

In [12], authors have indicated the priority assignment to the different robots in the dynamic system. The motivation behind this approach is to discover the issue of the path planning arrangements for different robots that can impart each other, to play out an arrangement of organized tasks in the dynamic condition. Path planning for the highly dynamic environment cannot work properly in this methodology.

In [13–17], researchers have expressed a portion of the calculations for organized path planning. Researchers proposed some of the methods such as prioritized planning (PP), revised prioritized planning (RPP), synchronous decentralized path planning (SDPP), asynchronous decentralized revised path planning (SDRPP), asynchronous decentralized path planning ADPP, and asynchronous decentralized revised path planning (ADRPP). Authors unmistakably separate between each of these calculations to demonstrate the working of the calculation effectively. But the priority assignment is fixed in this approach.

In [18], one uses DMA-RRT approach to design multi-agent teams in a cluttered environment. DMA-RRT is an extension of closed-loop RRT approach. This paper does not discuss priority assignment.

3 Motivation for This Work

3.1 Situation Where Prioritized Algorithm Works

Setting priority in the multiple robots is the key challenge nowadays. This may be of two kinds—static and dynamic priority schemes. In static priority scheme, priority assignment to the robots must take place at the initialization itself. One cannot change the priority of the robots in between once assigned. In dynamic priority scheme, initial priority assignment takes place at initialization. It can be used to change the priority according to the requirement. Preference will be given to the higher priority as compared to lower priority; otherwise, there must be chances of collision between the robots. All higher-priority robots avoid the path of lower-priority robots, and all lower-priority robots must change the path if they are in contact with higher-priority robots. Figures 1 and 2 show where one can set the priority to plan for motion in the multiple robots.

Fig. 1 Initial configuration

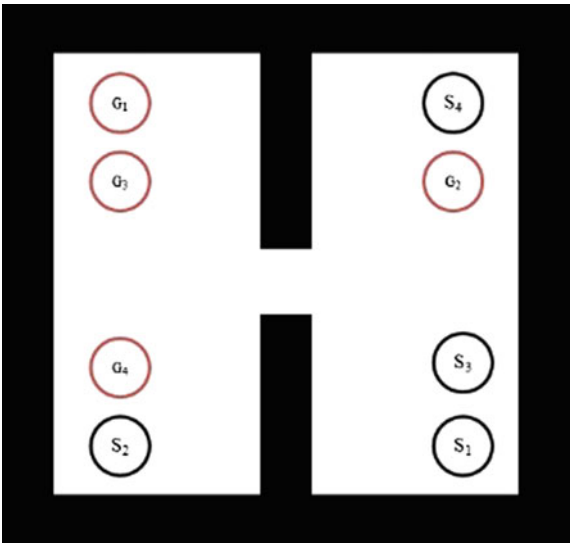
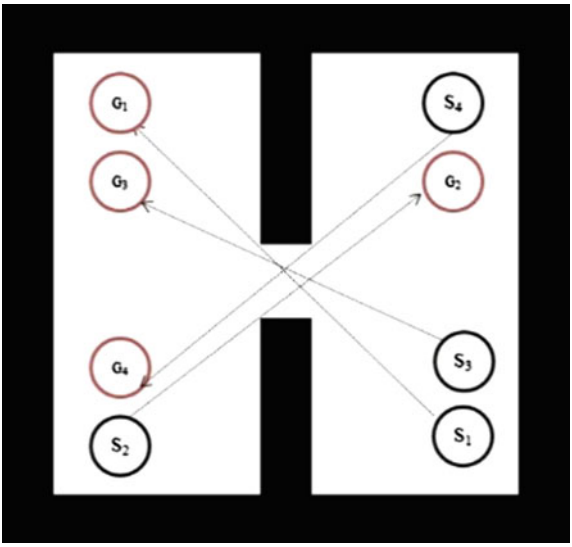


Fig. 2 Goal configuration



3.2 Situations Where Prioritized Algorithm Cannot Work

There are some situations where priority assignment may not work as this gives the same result—whether or not assign the priority to the robots. Figures 3 and 4 show the situation.

Fig. 3 Environment 1

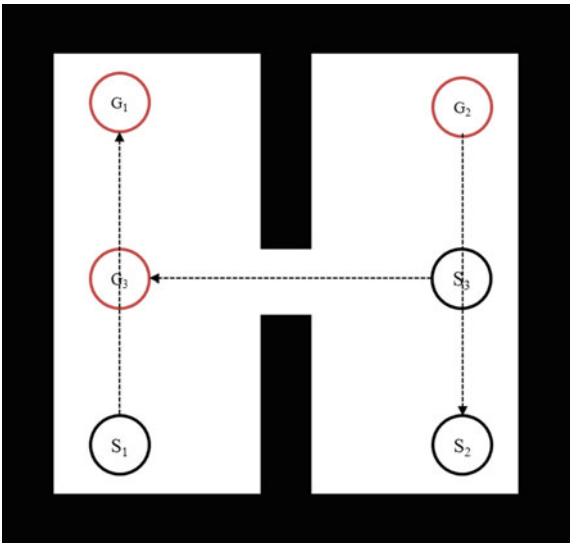
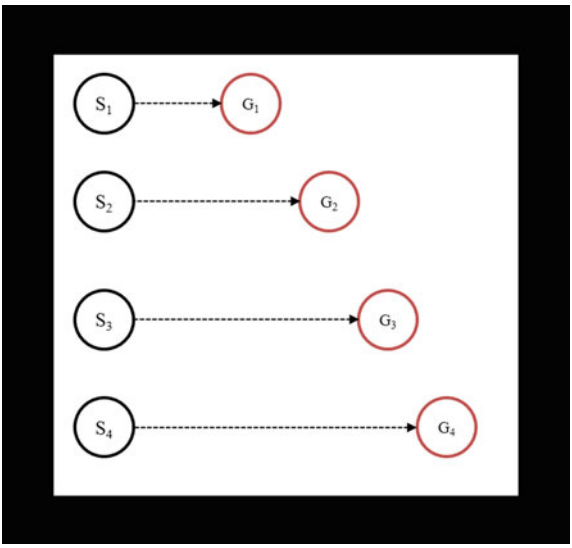


Fig. 4 Environment 2



4 Problem Model

4.1 Problem Formulation

Mathematically, multi-robot path planning can be characterized as follows: Consider a combination of m robots in 2D map, everyone with a given initial position and a respective goal position, and decide the path that every robot ought to take to achieve its objective, while keeping away from collisions with obstacles and different robots in the environment.

Assume that there are N homogeneous robots ($r_1, r_2, \ldots, r_n$) operating in a 2D map. Also, entire robots are isomorphic. One assumes additionally that there are M obstacles ($o_1, o_2, \ldots, o_m$) in the map and can be depicted as any unique geometry.

A path of robot r_i can be characterized as a set $p = \{(x_0, y_0), (x_1, y_1), \ldots, (x_m, y_m)\}$ where (x_j, y_j) is the position where robot must cover, and (x_0, y_0), (x_m, y_m) are the source position and destination position. Along these lines, the length of the way can be characterized as follows.

$$L_i = \sum_{i=1}^{m} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{1}$$

Multi-robot path planning aims to discover the shortest path and minimized cost that has collision-free maps consists of obstacles and robots.

4.2 Evaluation Criteria

In the motion planning approach, the fitness of the robots is measured in terms of path length and safety for evaluation which is utilized and appeared as follows:

(1) Path Length: Path length of the robot denoted as length (i) is described as given below.

$$length(i) = \sum_{i=1}^{n} |D_i| \tag{2}$$

where $Di = sqrt((x_i - x_{i+1})^2 + (y_i - y_{i+1})^2)$

(2) Path Safety: This is computed by the interspace difference across the robot and the obstacle. And this is described as given below.

$$safe(i) = \begin{cases} m, & \left[\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}\right] \leq d, \\ 0, & Otherwise \end{cases} \tag{3}$$

where (x_j, y_j) is the robot's position, (x_0, y_0) is the closest obstacle position, d is a safe distance, and m is a penalty.

4.3 Algorithm

```
Set initial position, goal position for n robots
Store the set of all priorities for n robots in priority-set
For priority-set=1 to n
   For i=1 to n
         Set the current position of x and y to initial position of x and y.
   End for
   For each robot i=1 to n
      While (current-i!=Goal-i)
         For j=1 to max_iteration
               Compute L_i
         End for
         If priority-i<priority-i+1
                  then slow the velocity of the i^th robot
         Else
               choose the alternate path to move
         End if
      End while
   End for
End for
For i=1 to n
      Evaluate the shortest paths of all the robots
End for
```

Table 1 Parameters

Parameter	Values
Map size	300 × 300
Team size	4, 5 and 6 robots
Sensor range	8 cell units
Max iteration	100
Robots initial location	Fixed at some points
Robots goal location	Fixed at some points

5 Experimental Setup

The prioritized multi-robot path planning is implemented in MATLAB *2013a* on *i5* processor with *2 GB* RAM. The simulation is performed with 4, 5, and 6 similar shaped robots. For each robot, the source and destination points are specified prior to executing the simulation. The results are shown by using brute force technique. Table 1 represents the different parameters for the environmental setup.

In our environment, some of the following assumptions have to follow for prioritized path planning:

1. Initially, assign the priority of the robots according to the initial robot numbers and find the path length.
2. In each iteration, change the priority of the robots and find the path length.
3. If lower-priority robot is moving on the path of higher-priority robot, then the velocity of the lower-priority robot must go down to avoid collision with higher-priority robot.
4. If the robots are moving in the environment and target position of another robot lies in the path of the robot, then treat that goal position as an obstacle if the robot has already available, otherwise treats it as a free path.

The following shows four different workspaces in which the simulations have been performed. All the robots were supposed to navigate from the fixed start position to the fixed goal positions. The algorithm runs on all four workspaces. The overall objective of the simulation is to calculate the shortest path for all the robots. Performance for the algorithm is calculated in terms of (a) movement steps of robots and (b) robot path length (Figs. 5, 6, 7, and 8).

6 Results

The result shows with respect to the given four workspaces. This part shows the success rate of the robots and respective path lengths. The results show the path length for all the combinations of priorities for the robots. Some of the results give lower path length, but there may be some situation where all the robots cannot reach

Fig. 5 Map 1

Fig. 6 Map 2

the destination. Some results give higher path length with all the robots that reach the destination (Tables 2, 3, and 4).

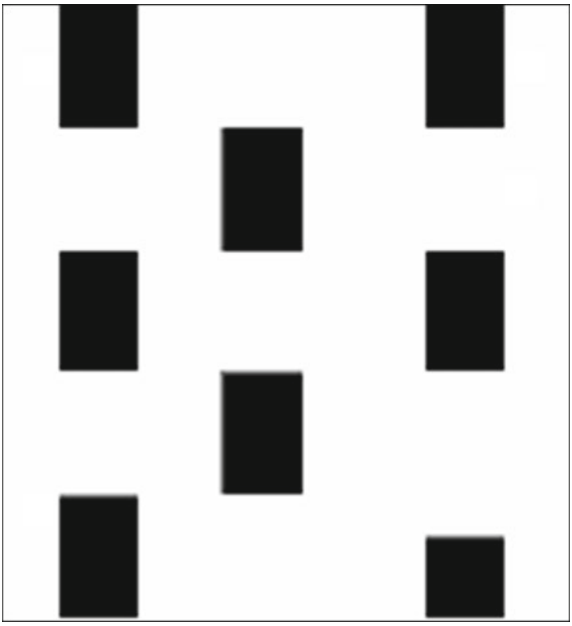


Fig. 7 Map 3

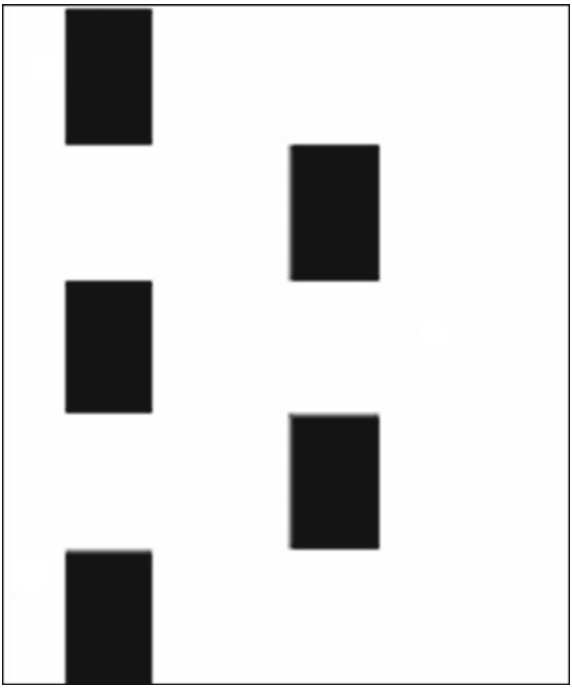


Fig. 8 Map 4

Table 2 Output for four robots

Map	Success rate	Path lengths	Minimum path length	Computation time	Priority set
Map 1	4	109	109	18.86	4, 3, 2, 1
Map 2	3/4	102/100/98/ 106/100/193	98	37.52	2, 3, 4, 1
Map 3	4	93/89	89	13.90	4, 3, 2, 1
Map 4	4	97/87	87	13.46	4, 3, 2, 1

Table 3 Output for five robots

Map	Success rate	Path lengths	Minimum path length	Computation time	Priority set
Map 1	5	109	109	24.02	5, 4, 3, 2, 1
Map 2	5	94	94	64.56	5, 4, 3, 2, 1
Map 3	5	104/100	100	18.43	5, 4, 3, 2, 1
Map 4	5	104/100	100	18.12	5, 4, 3, 2, 1

Table 4 Output for six robots

Map	Success rate	Path lengths	Minimum path length	Computation time	Priority set
Map 1	6	124	124	21.30	6, 5, 4, 3, 2, 1
Map 2	6	160	160	53.19	6, 5, 4, 3, 2, 1
Map 3	6	110/104	104	16.52	6, 5, 4, 3, 2, 1
Map 4	6	106/102	102	14.96	6, 5, 4, 3, 2, 1

7 Conclusion

This paper proposed a prioritized multi-robot path planning method that gives the brute force technique to find the optimized path length. Here, the main objective is to produce the shortest path length for all the robots. To do so, priorities of all the combinations of robots are proposed to minimize the time of computation. The efficiency is measured through multiple testing on environments by taking various evaluation parameters. This proposed method gives the optimum result as only one priority assignment leads sometimes to the worst result. Future work consists of environments that have heterogeneous robot size and also includes the condition wide range communication takes place.

References

1. Bennewitz M., Burgard W., Thrun S.: Optimizing schedules for prioritized path planning of multi-robot systems. In: Proceedings of IEEE International Conference on Robotics and Automation, Seoul, pp. 271–276 (2001)
2. Bennewitz M., Burgard W., Thrun S.: Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. Robot. Autom. Syst. pp. 89–99 (2002)
3. Bennewitz M., Burgard W., Thrun S.: Constraint-based optimization of priority schemes for decoupled path planning techniques. In: Proceedings of Annual Conference on Artificial Intelligence, pp. 78–93 (2001)
4. Bennewitz M., Burgard W., Thrun S.: Exploiting constraints during prioritized path planning for teams of mobile robots. In: Proceedings of International Conference on Intelligent Robots and Systems, Maul, hawaii, USA, pp. 393–398 (2001)
5. Yu W., Peng J., Zhang X.: A prioritized path planning algorithm for MMRS. In: Proceedings of 33rd Chinese Control Conference, Nanjing, China, pp. 966–971 (2014)
6. Yu W., Peng J., Zhang X. Lin K.: A cooperative path planning algorithm for a multiple mobile robot system in a dynamic environment. Int. J. Adv. Robot. Syst. pp. 1–12 (2014)
7. Berg J., Snoeyink J., Lin M., Manocha D.: Centralized path planning for multiple robots: optimal decoupling into sequential plans. In: Proceedings of International conference on Robotics: Science and Systems, University of Washington, Seattle, USA, pp. 1–8 (2009)
8. Frese C., Beyerer J.: A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles. IEEE Intell. Veh. Symp. pp. 1154–1160
9. Ma X., Jiao Z., Wang Z., Panagou D.: Decentralized prioritized motion planning for multiple autonomous UAVs in 3D polygonal obstacle environments. In: Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA USA, 2016, pp. 292–300
10. Clark C.M., Bretl T., Rock S.: Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems. In: Proceedings of the IEEE International Conference on Aerospace, 2002, pp. 3621–3631
11. Liu S., Sun D., Zhu C.: A dynamic priority based path planning for cooperation of multiple mobile robots in formation forming. Robot. Comput. Integr. Manuf. pp. 589–596 (2014)
12. Zheng T., Liu D.K., Wang P.: Priority based dynamic multiple robot path planning. In: Proceedings of 2nd International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, 2004, pp. 373–378
13. Cap M., Novak P., Kleiner A., Selecky M.: Prioritized planning algorithms for trajectory coordination of multiple mobile robots. IEEE Trans. Autom. Sci. Eng. pp. 835–849 (2015)
14. Cap M.: Algorithms for multi-robot trajectory planning in well-formed infrastructures. Assoc. Adv. Artif. Intell. pp. 1–5 (2015)
15. Berg J., Overmars M.: Prioritized motion planning for multiple robots. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, pp. 2217–2222
16. Cap M., Novak P., Vokrinek J., Pechoucek M.: Asynchronous decentralized algorithm for space time cooperative path finding. In: Spatio-Temporal Dynamics Workshop (SteDy), Proceedings of the European Conference on Artificial Intelligence, Montpellier, France, 2012, pp. 18–25
17. Cap M., Novak P., Selecky M., Faigl J., Vokrinek J.: Asynchronous decentralized prioritized planning for coordination in multi robot system. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 2013, pp. 3822–3829
18. Desaraju V.R., How J.P.: Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 4956–4961

Part III
Neural Networks

Modified Back Propagation Neural Network for Facial Expression Classification Using Principal Component Analysis and Ridgelet Transform

Ramachandran Vedantham, Lavanya Settipalli and Edara Sreenivasa Reddy

Abstract This article proposes a human emotion classification model using ridgelet transformation-based emotion image features and with back propagation artificial neural network. Ridgelet transformation method was applied to face images around the points of interest area. Principal component analysis technique was used for preprocessing phase in feature extraction from emotion-contained images. The proposed model has two important phases that are finely mixed to make the entire system optimal. The first step is to select the image representation that extracts the important features of image using PCA and then apply ridgelet transform to get the fine-tuned features from the obtained eigenfeatures. The second step is a meticulously modified back propagation artificial neural network classifier which includes two parameters such as anxiety and confidence, for selecting a subset of typical conclusive visual topographies from the set of latent structures. The combination of these two procedures enabled rapid learning and thereby computational swiftness reasoning the auspicious facial emotions.

Keywords Human emotion classification · PCA · Ridgelet transforms
Back propagation neural network

R. Vedantham (✉) · E. S. Reddy
Acharya Nagarjuna University, Guntur, Andhra Pradesh, India
e-mail: vrc.bhatt@gmail.com

E. S. Reddy
e-mail: edara_67@yahoo.com

L. Settipalli
Vignan's University, Guntur, Andhra Pradesh, India
e-mail: spl.lavanya86@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_14

1 Introduction

Facial emotion conveys non-spoken communiqué cues. The investigation on automatic emotion identification is growing gradually because of its basic applications, for example, smart home arrangements, cooperating calling systems, intellectual written course classes. In a home with smart arrangements, contingent on the feeling of an occupant, a little preferred music can mechanically be played to transform the occupant's minus to a plus feeling. In an interactive call arrangement, the reply can be altered by conferring to the feeling of clienteles. In an intellectual spoken course class, knowing and familiarizing to students' feelings is a significant component to decrease the slit of performance among manual and automatic classes.

Robotic emotions identification is not a new subject but the way to handle it varies with the researcher's keen interest in which he deals with various parameters which are effective robotic emotions; the existing methods simply stresses on one way of the method. Spectral, prosodic, and cepstral are the commonly observed things in speech modality to notify that the emotions, both verbal and audio features, [1] are being combined together. In SVM methods, many of the authors have stressed on hybrid classifier which is a support vector [2]. Out of all the parameters being examined, it was traced out that prosodic features were very helpful in increasing the pitch period [3]; with the help of a genetic algorithm related to fuzzy neural network along with a required classifier, this can be achieved as discussed in [4]. Besides, these many dissimilar classifiers which were hierarchical in natures were used in various emotion detections as discussed in [5]; all the results associated with these parameters are associated with a common database [6].

In this article, we propose a classification model for facial emotions with modified back propagation neural network [7, 8] that uses principal component analysis technique [9] for preprocessing phase in feature extraction of face emotion images, and then ridgelet transformation is applied on the extracted features from PCA [10]. This method is capable of processing images enormously rapid while achieving high recognition rates for facial emotions. This whole system deploys two significant contributions that are optimally mixed to make the whole system more rapid. The first step is to select an image representation that extracts the important features from image using PCA and then apply ridgelet transform to get the fine-tuned features from the obtained eigenfeatures. The second step is a scrupulously modified back propagation neural network classifier which includes two parameters such as anxiety and confidence, which is used to choose a subset of typical decisive visual features out of a set of potential features.

1.1 Ridgelet Transform

The ridgelet transformation discovered kind of anisotropic geometric wavelets transform [11–13]. The general wavelets fail abruptly in order to signify the objects with

high-valued anisotropic 2D-elements, such as curvilinear or line structures. The final results pointed out those non-geometrical wavelets are generated and may not exploit the streamlining of edge curve. These transformations may lead to the perfect peculiarities which can be categorized into straight lines. The results of these transform analyses lead to the formation of a one-dimensional wavelet with a radon domain which can be used as a rudimentary tool for calculating ridgelet coefficients. The radon transforms.

$$\mathrm{R}: L2(R2) \rightarrow L2([0, 2\pi], L2(R)) \tag{1}$$

are defined by equation

$$Rf(\theta, t) = f(x1, x2)\delta(x1 \cos\theta + x2 \sin\theta - t)dx1\, dx2 \tag{2}$$

Here (x1, x2) is the position vector, δ is the Dirac delta function, and $x1 \cdot cos\,\theta + x2 \cdot sin\,\theta$ is called the ridge. Here, f is described with the examination of the radon transform via equation, in R*f* (a, b, θ) of an entity

$$Rf(a, b, \theta) = Rf(\theta, t)a - \frac{1}{2}\psi\left(\frac{(t-b)}{a}\right)dt \tag{3}$$

Here, wavelet is having two parameters a and b, and ψ is considered as the mother wavelet. One-dimensional wavelet transform represents an exact ridgelet transformation which is proportional to the radon transform, with t as a variable, and which is a variable with angular change, and the constant term is θ [12–14].

An initial approach in tracing the ridgelet transformation is to calculate the random transformation *Rf* (*θ, t*) as the first step and to apply a 1D wavelet transformation to the portions *Rf* (*θ, t*) as the second step. Applying the 1D inverse Fourier transform to the 2D Fourier transform is restricted to radial lines through the origin [12, 15] which will yield radon transformation. To show how a digital data is dependent on FET, these considerations are assumed to be enough.

The steps to calculate these are [16]:

1. Find the 2D FFT of *f*.
2. With the help of an interpolation scheme, the generation of polar from Cartesians is carried out, and "f" the sampled values obtained on a polar lattice are very helpful in replacing the Fourier transformation having experimental values. Calculate the 1D IFFT on each line.
3. Perform ridgelet transformation, as mentioned earlier, on each standardized partition.
4. Find the ridgelet coefficients.

Ridgelet transforms basic flow is presented in Fig. 1. With the help of the prior described cuboids, ridgelet coefficients are being obtained.

A sentimental analysis in the recent years was proposed by Poria et al. For this analysis, they used audio, textual and visual information, and, as discussed by Hossain

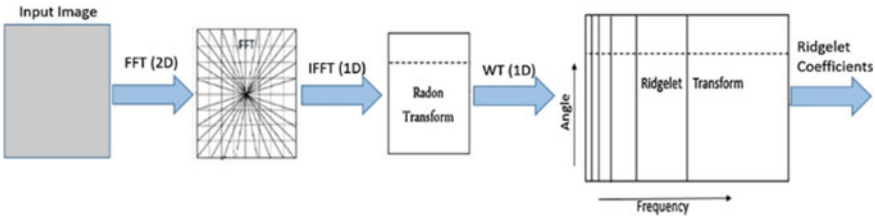


Fig. 1 Basic flow of ridgelet transform

and Muhammad for the detection of a patient status, they used both video-audio as input. In contrast to the methods, we are not applying any textual or temporal domain with regression along with frequency regression.

1.2 Emotional Back Propagation Neural Network Algorithm

BP in neural networks is also known as widespread delta rule as suggested by Rumelhart et al. [9, 17–20]. With an ease employment and rapid training, this algorithm is frequently being implemented with a proposed EMBP, whose details are elucidated as per the flow of relevant data in the neural network, where it entails three layers: input layer with i neurons h neurons, hidden layer, and output layer with j neurons.

1.2.1 Input-Layer Neurons

Input-layer neurons are not dispensation, and thus, the production of every input-layer neuron is defined as

$$YI_i = XI_i \tag{4}$$

where YI_{i} and XI_{i} are in the order of input and output values of neuron i in the input layer. Figure 2 resembles the input and output formation of a neuron in the input layer.

1.2.2 Hidden-Layer Neurons

These kind of neurons are processing-type neurons, and hence, the sigmoid initiation method is used to trigger every neuron in this layer. We assume there will be one hidden layer; even if there is more, the similar steps are useful to accommodate an additional hidden layer using the following algorithm.

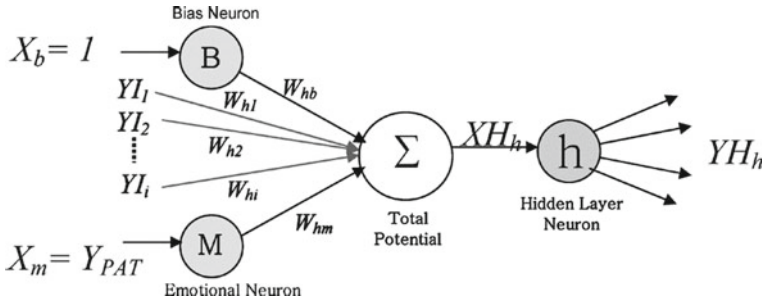


Fig. 2 I/O configuration of a hidden-layer neuron

The hidden-layer neuron values are calculated as follows

$$YH_h = \left(\frac{1}{1 + exp(-XH_h)} \right) \tag{5}$$

where XH_h is the input value and YH_h is the output value of neuron in the hidden layer.

The XH_h, which is input of a hidden-layer neuron, is premeditated from the total potential TP_h of all input values that enter into that neuron. Total potential is defined as the amount of growths of input values along with their associated weights.

Three different types of inputs exist with respect to three types of total potential, feeding into a hidden-layer neuron. TP_{hc} is first total potential, which is presumed as conventional total potential, where it is gained using the output values of the earlier layer and weight matrix. The second total potential TP_{hc} is gained from the hidden-layer bias neuron along with its related weights. TP_{hm}, which is the third total potential, is gained using the hidden-layer emotional neuron and corresponding weights.

TP_{hm} is defined as

$$TP_{hm} = W_{hm} \cdot X_m \tag{6}$$

TP_{hc} is calculated as

$$TP_{hc} = \sum_{i=1}^{r} W_{hi} \cdot YI_i \tag{7}$$

Here, W_{hi} is weight denoted by hidden h to input i neuron, and YI_i is assumed to be output of input i neuron, and r is the highest number of input neurons.

1.2.3 Output-Layer Neurons

There exist dispensation neurons too, which exhibit similar nature as concealed layer; the sigmoid, which is considered as the activation function, is used to activate every neuron in the layer. The final outcome of every outcome layer is described as

$$YJ_i = \left(\frac{1}{1 + exp(-XJ_i)} \right) \tag{8}$$

The output and input values of a neuron are considered as YH_h and XH_h, respectively. So, the input-to-output-layer neuron is calculated as follows

$$XJ_j = TP_{jc} + TP_{jb} + TP_{jm} \tag{9}$$

TP_{jc} is defined as

$$TP_{jc} = \sum_{h=1}^{l} W_{jh} \cdot YH_h \tag{10}$$

where W_{jh} is called as weight, which is given by output-hidden neuron, and YH_h is called the output, which is obtained from hidden neuron, and l is considered as the highest number of hidden-layer neurons.

TP_{jb} is calculated as

$$TPjm = Whm \cdot Xm \tag{11}$$

Here, Wjm is considered as the weight, which is obtained by output-to-output-layer emotional neuron m, and Xm is considered as the input to the emotional neuron. Xm is obtained as in (7). Figure 3 depicts configuration of the input and output of an output-layer neuron.

1.2.4 The Emotional Back Propagation Parameters

The mentioned emotional parameters are utilized along with respect to the pre-existing learning coefficient (η) and momentum rate (α) for adjusting error minimization with respect to the neural network weights.

We have stressed to bring up two parameters: confidence coefficient (k) and anxiety coefficient (μ), which we have analyzed that while learning a new task, it was observed that the levels of anxiety are very high at the starting and a low confidence level. As the time passes and in practice, it was observed that quite converse is happening; such as anxiety level comes down while the level of confidence increases. As the iterations progress, the learning becomes reaches completion, leading to lessened anxiety and increased confidence for learning rest of the work.

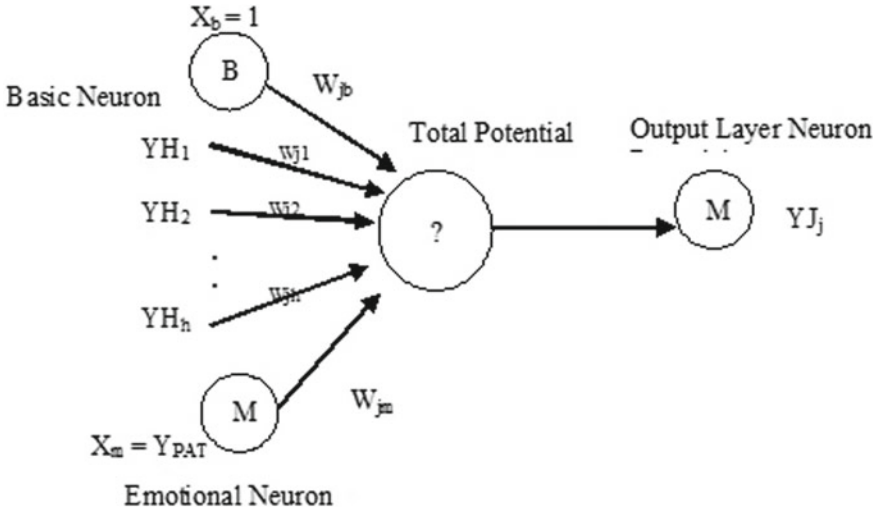


Fig. 3 I/O configuration of an output-layer neuron. *Courtesy* Adnan Khashman (2008)

The mentioned emotional back propagation neural network uses both the generalization processes and emotional coefficients during the learning. The two above-said coefficients are having value between "0" and "1". For modeling the emotional factor with respect to the ANN, we have assumed the following conditions below.

Condition 1. Input parameters are dependent on the anxiety levels. Higher anxiety is caused by the new patterns. Initial anxiety level is assumed as 1 during first iteration of the concept.

Condition 2. Between neural network and actual output parameter, the anxiety level is considered as to be dependent.

Condition 3. With a decrease in the level of anxiety, level of confidence increases; on the above-specified condition, the anxiety parameter is calculated as

$$\mu = Y_{AVPAT} + E \tag{12}$$

Here Y_{AvPAT} is called as mean value of all patterns fed to the NN in every epoch and is calculated as

$$Y_{AVPAT} = \frac{\sum_{P=1}^{NP} Y_{PAT}}{N} \tag{13}$$

Here, p is called as index of each pattern from first to last. $YPAT$ is the mean value of pattern. Feedback error E is calculated as

$$E = \sum_{j=1}^{N_j} (Tj - YJj)^2 / Np \cdot Nj \tag{14}$$

Here *j* is called as index of output neuron from first to last and is the total number of patterns. YJj and Tj are the original and target output value for neuron j, respectively.

The confidence parameter (κ) value is calculated as

$$\kappa = \mu\, 0 - \mu\, i \tag{15}$$

Here, $\mu 0$ is the anxiety parameter after the first epoch and μi is the anxiety parameter at following epoch i.

2 Proposed Methodology

Figure 3 depicts the architecture of facial emotion recognition model, which is being proposed. The first step in the proposed system is the preprocessing step. Face position is considered as the important data that is needed for most of the facial emotion recognition method. In preprocessing phase, face images are cropped from 256 × 256 pixels to 100 × 100 pixels-sized images (Fig. 4).

There are five steps involved in the proposed architecture: preprocessing of given images, extraction feature, and classification, training phase and knowledge construction. The classification task is achieved by feeding features obtained from the testing image and follows a single pass.

2.1 Databases

The standard dataset called eNTERFACE dataset [21, 22] is used in experimentation. Figure 5 depicts samples' expressions of the considered dataset. A small experiment was carried out in order to trace out the various facial expressions which appear naturally on every person's face depending on the situations; to carry out this, a specific person was selected and he/she was made to listen six different stories which were full of different emotions, and while listening a particular story, the experimental facial expressions were captured; it was clearly observed that all the six story-listening expressions were not of the same, and thus our required content has been bubbled out.

In order to compare with other datasets, Kanade–Cohn dataset is also used. The persons posing for expression images in Kanade–Cohn [9, 17–20] dataset are trained for the job before capturing the images.

Facial expressions from the image database of Cohn–Kanade dataset are used. Sample images are shown in Fig. 6. The original face images contain time and camera model also. For convenience and better accuracy, face region is identified, cropped, and stored separately. The facial emotion image, which is cropped, is used to obtain distinguishable features. Those obtained features then fed the ANN, and ANN will be trained to construct knowledge database.

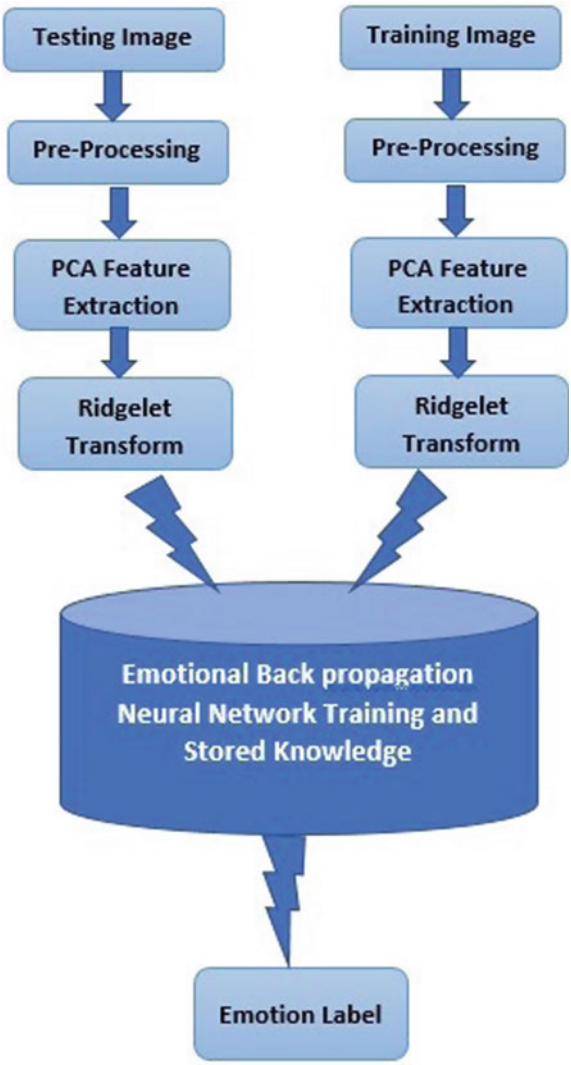


Fig. 4 Architecture of proposed model

3 Tests, Results, and Discussion

The accuracy of the proposed model is recorded by changing the number of images to the training as well as testing. Table 1 depicts the average emotion-specific accuracies of the proposed model. The accuracies increase with the increase in the number of training images. It is understood that the ridgelet transform along PCA and EMBP-NN is producing very good accuracies despite less number of samples.

The Table 1 below represents the proposed system's confusion matrix, by using eNTERFACE. From the confusion matrix it can be observed that, the emotions

Fig. 5 Sample images of eNTERFACE database

Fig. 6 Sample images of Cohn–Kanade database

Table 1 Confusion matrix on eNTERFACE dataset

	Surprise	Fear	Happy	Sad	Anger	Disgust
Surprise	82.5	2.5	2.3	2.5	5.2	5
Fear	2.9	74.6	2.1	7.4	5.1	7.9
Happy	3.2	4.2	85.6	2.4	2.4	2.2
Sad	3.6	5.2	3.2	76.3	4.4	7.3
Anger	5.6	4.2	5.1	6.2	73.4	8.2
Disgust	11.2	4.3	12.1	8.3	4.2	81.5

Surprise, Happy and Disgust were having satisfactory values, while Fear, Sad and Anger registered low values.

Table 2 presents confusion matrix by consideration of various modality from Kanade–Cohn database. The average accuracy is around 84.16%; the emotion accuracy was observed to be higher which is related to Kanade–Cohn databases which are being well trained.

Table 2 Confusion matrix on Cohn–Kanade dataset

	Surprise	Fear	Happy	Sad	Anger	Disgust
Surprise	88.5	2.9	3.2	3.6	5.2	3.2
Fear	2.5	78.6	4.2	5.6	4.2	5.1
Happy	2.3	2.1	82.2	11.2	4.3	12.1
Sad	2.5	5.2	5	74.6	6.2	8.3
Anger	7.4	5.1	7.9	4.4	73.6	4.2
Disgust	2.4	2.4	2.2	7.3	8.2	81.3

Table 3 Accuracy comparison with existing models on eNTERFACE dataset

System	Features	Classifier	Accuracy (%)
Rumelhart et al. [20]	ITMI, QIM, MFCC, prosody, formants	Multi-classifier	77.78
Murase and Nayar [21]	Prosody and facial markers	SVM	71
Chavan et al. [8]	MFCC, prosody, geometric markers	TSD Bayesian	66.54
Gosavi and Khot [10]	MFCC, facial markers coefficients, linear prediction prosody	SVM, ANN	39
Proposed	PCA, Ridgelet coefficients	EBPNN	84.16

The confusion matrix portrays the proportion of accurate divisions against wrong guess. The diagonal values represent the correctly guessed emotions. All the other values are wrong guesses. Experimental outcomes show that the anticipated methodology enhances the presentation of the facial emotion classification. Based on the obtained consequences, we can say that the anticipated neural network with emotional back propagation with ridgelet transform and PCA is outperforming in both cases of reducing the training time and increasing the accuracy (Tables 3 and 4).

Momentum and learning coefficient play a major role for the performance of a neural network. On consideration of value 0.02 related to recognition of facial emotions and thus leads to best performance, this can be achieved with the help of neural networks, for getting excellent performance along with less timing, segmentations like clustering techniques are being considered, which may vary from person to person.

4 Conclusion and Future Work

A novel approach for facial emotion recognition is proposed using ridgelet transform, principal component analysis, and emotional back propagation neural networks in this paper. The system was evaluated in eNTERFACE and Cohn–Kanade datasets.

Table 4 Accuracy comparison with existing methods on Cohn–Kanade dataset

System	Features	Classifier	Accuracy (%)
Rumelhart et al. [20]	ITMI, QIM, MFCC, prosody, formants	Multi-classifier	78.68
Murase and Nayar [21]	Prosody and facial markers	SVM	73.56
Chavan et al. [8]	MFCC, prosody, geometric markers	TSD Bayesian	68.34
Gosavi and Khot [10]	MFCC, facial markers coefficients, linear prediction prosody	SVM, ANN	45.43
Proposed	PCA, ridgelet coefficients	EBPNN	89.27

The accuracies of 81.04 and 84.16% were obtained. These obtained accuracies are much better than some of the existing models specified in the literature on the same database. Fear and Happy emotions recorded accuracies less than 80%, so we can concentrate more on these emotions to improve the accuracies.

The design of classification algorithm is carried out to take facial expressions with temporal dynamics. For a variety of facial expressions, our future work reflects the presentation of the classifier with a training data in order to obtain various publically available databases along with the reflecting broader sense.

References

1. Schuller B., Rigoll G., Lang M.: Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine—belief network architecture. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 577–580 (2004)
2. Zhou Y., Sun Y., Zhang J., Yan Y.: Speech emotion recognition using both spectral and prosodic features. In: Proceedings of International Conference Information Engineering and Computer Science (ICIECS), pp. 1–4 (2009)
3. Devillers L., Vidrascu V.: Real-life emotion detection with lexical and paralinguistic cues on Human-Human call center dialogs. In: Proceedings of Interspeech'2006, Pittsburg (2006)
4. Gharavian, D., Sheikhan, M., Nazerieh, A.R., Garoucy, S.: Speech emotion recognition using FCBF feature selection method and GA optimized fuzzy ARTMAP neural network. Neural Comput. Appl. **21**(8), 2115–2126 (2012). https://doi.org/10.1007/s00521-011-0643-1
5. Albornoz, E.M., Milone, D.H., Rufiner, H.L.: Spoken emotion recognition using hierarchical classifiers. Comput. Speech Lang. **25**(3), 556–570 (2011)
6. Burkhardt F, Paeschke A, Rolfes M, Sendlmeier W, Weiss B: A database of German emotional speech. In: Proceedings of Interspeech'2005, Lisbon (2005)
7. Adnan Khashman: A modified back propagation learning algorithm with added emotional coefficients. IEEE Trans. Neural Netw. **19**(11) (2008)
8. Chavan P.M., Jadhav, M.C., Mashruwala, J.B., Nehete, A.K., Panjari, P.A.: Real time emotion recognition through facial emotions for desktop devices. Int. J. Emerg. Sci. Eng. (IJESE) **1**(7) (2013)

9. Ramachandran, V., Sivaiah, B., Srinivasa, R.E.: Facial expression classification system with emotional back propagation neural network. Int. J. Sci. Eng. Res. **4**(10), 383–391 (2013)
10. Gosavi, A.P., Khot S.R.: Facial emotion recognition using principal component analysis. Int. J. Soft Comput. Eng. (IJSCE) **3**(4) (2013)
11. Do, M.N., Vetterli, M.: The finite ridgelet transform for image representation. IEEE Trans. Image Process. **12**(1), 16–28 (2003)
12. Bejani, M., Gharavian, D., Charkari, N.M.: Audiovisual emotion recognition using ANOVA feature selection method and multiclassifier neural networks. Neural Comput. Appl. **24**(2), 399–412 (2014)
13. Mansoorizadeh, M., Charkari, N.M.: Multimodal information fusion application to human emotion recognition from face and speech. Multimed Tools Appl. **49**(2), 277–297 (2010)
14. Jiang, D., Cui, Y., Zhang, X., Fan, P., Ganzalez, I., Sahli, H.: Audio visual emotion recognition based on triple-stream dynamic bayesian network models. In: D'Mello S. et al. (ed.) ACII 2011, Part I, LNCS 6974, pp. 609–618 (2011)
15. Paleari, M., Huet, B.: Toward emotion indexing of multimedia excerpts. In: Proceedings of International Workshop on Content Based Multimedia Indexing (CBMI), London, pp. 425–432 (2008)
16. Baboo, S.S., Baboo, S.S.: Face emotion analysis using gabor features. In: Image Database for Crime Investigation. Int. J. Data Eng. (IJDE) **2**(2) (2011)
17. Sivaiah, B., Vamsidhar, T., Chandana, K.H.: An efficient approach for fingerprint recognition by matching minutiae pairings. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **5**(2), 536–542 (2015)
18. Ramachandran, V., Srinivasa Reddy, E., Sivaiah, B.: A novel facial expression classification system using emotional back propagation artificial neural network using genetic algorithm. Int. J. Appl. Eng. Res. **10**, 38583–38588 (2015). ISSN 0973-4562
19. Ramachandran, V., Srinivasa Reddy, E., Sivaiah, B.: An enhanced facial expression classification system using emotional back propagation artificial neural network with DCT approach. Int. J. Comput. Sci. Eng. Inf. Technol. Res. (IJCSEITR) **5**, 83–94 (2015). ISSN(P): 2249-6831; ISSN(E): 2249-7943
20. Rumelhart, D., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D., McClelland, J. (eds.) Parallel Distributed Processing, vol. 1. MIT Press, Cambridge, MA (1986)
21. Murase, H., Nayar, S.K.: Visual learning and recognition of 3-D objects from appearance. Int. J. Comput. Vis. **14**, 5–24 (1995)
22. Li, S.Z., Hou, X.W., Zhang, H.J., Cheng, Q.S.: Learning spatially localized, parts-based representation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 207–212 (2001)

Classification of Hand Movement Stages for Brain–Computer Interface Using Convolutional Neural Network

Kriti Singhal, Evi Agarwal, Arpit Yadav and Anuraj Singh

Abstract In this paper, a deep learning based control algorithm for a fully functional, real-time prosthetic limb is presented to provide motor rehabilitation and restoration. The proposed scheme detects six different events related to hand movement during a task of grasping and lifting an object, using electroencephalographs (EEG). These six events represent the six sequential stages of a grasp-and-lift action like hand starts moving, starts lifting the object. This method mainly aims at increasing the classification performance of BCI applications by incorporating convolutional neural network (CNN) and low-pass filtering based learning approaches on electroencephalography recordings.

Keywords BCI · Electroencephalography · Convolutional neural network Grasp-and-lift task · Adadelta · AUROC

1 Introduction

Mobility rehabilitation can be defined as a type of physical rehabilitation used for patients who are suffering from mobility issues, caused by some accidents or physical disabilities. People suffering from such disabilities are underprivileged of voluntary control of body parts and need to be assisted by prosthetic devices. Brain–computer interfaces (BCIs) are very popular topics of research nowadays in the field of prosthetic control devices to restore lost motor functions [2]. BCIs connect human brain

K. Singhal (✉) · E. Agarwal · A. Yadav · A. Singh
ABV-Indian Institute of Information Technology & Management, Gwalior, India
e-mail: kritisinghal2211@gmail.com

E. Agarwal
e-mail: evi.agarwal@gmail.com

A. Yadav
e-mail: yadavarpitbcn@gmail.com

A. Singh
e-mail: anurajiitr@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_15

Fig. 1 Complete processing pipeline of the proposed model

and outside world by acting as a bridge between them to facilitate mind controlling of machines using noninvasive brain signals. BCIs have emerged out as affordable, user-friendly, efficient, and portable devices [12].

Such BCI devices are mostly built upon electroencephalography (EEGs) which refers to the recording of the electrical activity of the brain in the form of waveforms which in turn reflects the cortical electrical activity. Electroencephalographs give a broad view of neural activity and have been used to study the physiology of brain in a noninvasive manner for years. Past studies have revealed the usage of EEG signals to even differentiate the movement of individual fingers [6].

A wide variety of machine learning-based approaches have already been undertaken in neuroscience in order to obtain a deeper knowledge about the human brain. In the previous study, a comparative analysis of three different machine learning-based approaches for hand movement detection, namely normalization with convolutional neural network (CNN), logistic regression with band-pass filter bank common spatial pattern, and with low-pass filter bank, has been made [11]. Among all three, CNN-based approach achieved the best AUROC value of 0.829. Another study in [3] illustrated five different RNN architectures for identifying hand movement stages of a task involving grasping and lifting an object.

This paper proposes a method for identifying six different stages of hand motion constituting a task of grasping and lifting an object from EEG recordings using CNN and Butterworth low-pass filter. Firstly, the raw EEG signals are preprocessed and normalized and then fed into convolutional neural network which finally outputs the probability of each of the six different events. The processing pipeline of the proposed algorithm is shown in Fig. 1.

2 Dataset and Preprocessing of Data

Raw EEG data cannot be used for the designated task of predicting hand movement stages through a BCI application; it needs to be preprocessed according to the requirement so as to make it easy to manipulate, and analysis of the data becomes easier.

2.1 Dataset

The dataset is collected from WAY-EEG-GAL which is formulated to decode and analyze EEG signals in humans who have performed the task of grasping and lifting an object [8]. It consists of EEG recordings of grasp-and-lift tasks from 12 subject in total, where each subject performed ten series of trials. A series comprised of 30 trials approximately. EEG recordings of each trial are from 32 channels in a standard configuration. The spatial relationship between the electrode locations is taken from [5] and shown in Fig. 2. Each trial is split into six phases: HandStart, FirstDigitTouch, BothStartLoadPhase, LiftOff, Replace, and BothReleased. These six events and their positions on the input EEG signals are graphically illustrated in Fig. 3.

The task of grasping and lifting an object is defined as moving the hand toward an object, then lifting it with thumb and index finger, holding it, and then placing it at a different position. The subjects performing the experiment were signaled to reach for the object and lowering it down by turning on and turning off the LED light

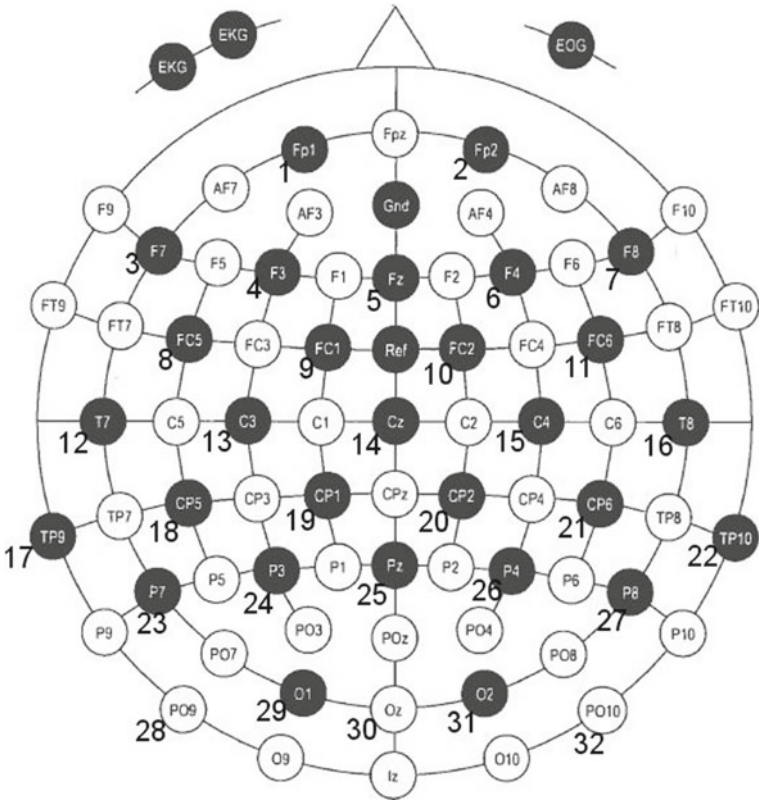


Fig. 2 Scheme of BCI electrode spatial locations [5]

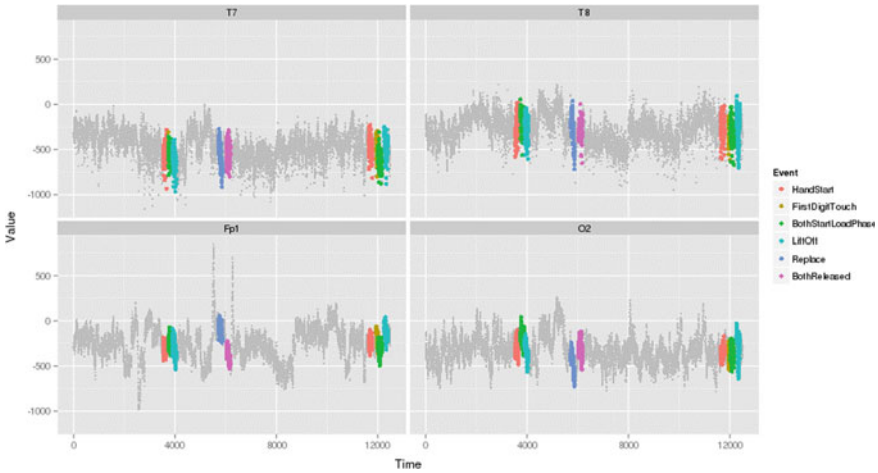


Fig. 3 Illustration of the six hand movement stages on raw EEG signals

Table 1 Six stages of hand movement and their description

Events	Description
HandStart	Starting hand movement to reach for the object
FirstDigitTouch	Grasping the object
BothStartLoadPhase	Onset of the load phase
LiftOff	Lifting the object and holding it for few seconds
Replace	Placing the object at a different location on the surface
BothReleased	Releasing the object

respectively. Approximately 328 trials were performed by each participant with a sampling rate of 500 Hz. This means 500 time frames of 0.002 s are recorded each second. The six events are described in Table 1.

2.2 *Preprocessing of Data*

The biggest challenge with monitoring EEG is artifact recognition and elimination. There are patient-related artifacts like sweating, eye movements that can be present in the raw EEG signals [9]. The front polar lobe electrode Fp1 is generally known to record eye blinks along with the EEG and hence acts as the main cause for artifacts. Therefore, the artifact rejection is performed only on Fp1 electrode by applying a third-order Butterworth low-pass filter [7].

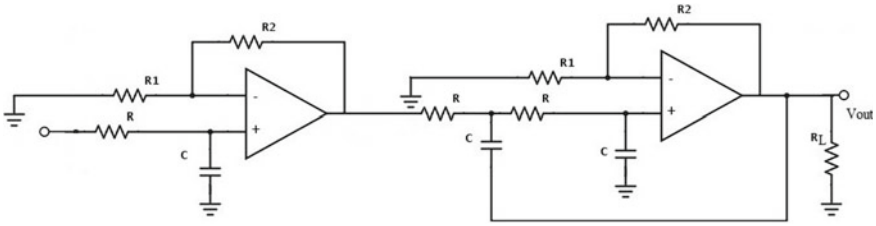


Fig. 4 Circuit diagram of a third-order Butterworth filter [1]

If n is the number of poles that a circuit has, f is the operating frequency, and f_c is the cutoff frequency of the circuit, then the Butterworth filter is considered a nth-order filter and its amplitude response is given by the formula

$$\frac{V_{out}}{V_{in}} = \frac{1}{\sqrt{(1 + (\frac{f}{f_c})^{2n})}} \tag{1}$$

The circuit of the third-order Butterworth filter is shown in Fig. 4.

In total, ten Butterworth low-pass filters of order 3 are applied on the Fp1 electrode leading to an increase in a number of electrode channels from 32 to 52. Since Butterworth filters require the cutoff frequency to be normalized, the cutoff frequency is divided by the Nyquist frequency (half of sampling rate) which is 250 Hz according to the dataset used. After applying low-pass filtering, the signals are normalized using mean and standard deviation estimates calculated from the dataset.

$$x_{i_{new}} = \frac{(x_{i_{old}} - \mu_x)}{\sigma_x} \tag{2}$$

where

x_i elements of column X.
X columns of the data set matrix.
μ_x mean of the column X.
σ_x standard deviation of column X.

3 Classification Technique: Convolutional Neural Network

The main principle is to develop a model, which is capable of extracting the important features from the electroencephalographs without significant data preprocessing or a priori information. The paper describes one-dimensional convolutional neural

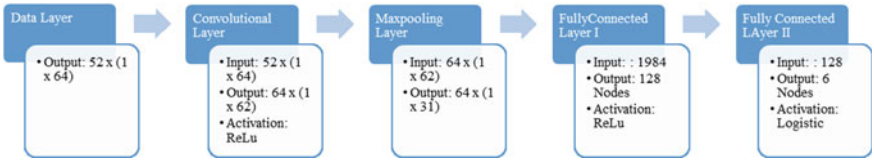


Fig. 5 Complete architecture of the convolutional neural network

network (CNN) that is capable of learning the activation patterns of electroencephalographs, by applying convolution only on time axis instead of frequency and spatial axis [10]. The input to the network is the raw EEG signals from 52 channels, sampled at the rate of 500 Hz.

3.1 CNN Structure

The CNN constitutes of five layers, namely data layer, 1D convolutional layer, max-pooling layer, and two fully connected layers shown in Fig. 5. Data layer is the first layer of the network which connects the dataset to the network. After low-pass filtering, the 32 electrode channels get converted into 52 channels. A time window of 1024 samples is taken which is subsampled by picking every 16th sample to create an observation consisting of 64 samples. The data layer then feeds these 52 (1 × 64) vectors into the convolutional layer.

The convolutional layer uses 64 filters with each filter vector of size 1 × 3 resulting in 64 output feature maps of size 1 × 62. This layer performs 1D convolution along the time dimension. The activation function used is rectified linear unit (ReLU) [14] given by the formula

$$f(x) = max(x, 0) \tag{3}$$

The max-pooling layer also known as downsampling layer makes the network more robust against feature offsets. A filter of shape 1 × 2 with a stride of 2 is used to compress the input feature maps by a factor of 0.5. This layer is followed by two consecutive fully connected layers. The first one has 1984 inputs and 128 outputs with the activation function as ReLU. The second layer combines the previous fully connected layers output values to calculate the probabilities of each of the six events. It uses logistic/sigmoid activation function to generate the output probability values given by the formula

$$f(x) = \frac{1}{(1 + exp(-x))} \tag{4}$$

3.2 Optimization Technique

In the model, cross-entropy function is used as the cost function. If there are total n tuples in the training data, x is input, and y is the desired corresponding output, then taking sum over all the training input points, cross-entropy cost function can be defined as

$$C = -\frac{1}{n}\sum_{x}[ylna + (1-y)ln(1-a)] \tag{5}$$

In the model, Adadelta [13] is used to optimize the cross-entropy function because in this technique there is no need of manually tuning the learning rate.
Update rule for Adadelta is as follows

1. Computation of gradient g_t at current time t.
2. Accumulation of gradients

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho)g_t^2 \tag{6}$$

3. Computing update

$$\Delta x_t = -\frac{\sqrt{E[\Delta x^2]_{t-1} + \epsilon}}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{7}$$

4. Accumulating update

$$E[\Delta x^2]_t = \rho E[\Delta x^2]_{t-1} + (1-\rho)\Delta x_t^2 \tag{8}$$

5. Applying update

$$x_{t+1} = x_t + \Delta x_t \tag{9}$$

where ρ is a decay constant and ϵ is for numerical stability.

Adadelta appears robust to noisy gradient information, different data modalities, and selection of hyperparameters. A separate dynamic learning rate per dimension is used in this technique, and it requires minimal computation over other gradient descent optimization techniques. The main advantage of using Adadelta is that it scales the learning rate based on recent historical gradient.

3.3 Performance Estimation

The performance metric used for the evaluation of the model is the area under the ROC curve (AUROC) [4]. ROC curve is a plot between the true positive rate (TPR) and false positive rate (FPR) for different threshold values.

True Positive Rate: TPR is defined as the proportion of positive samples (in our case the target event among the six events) that are identified as positive by the classifier.

$$TPR = \frac{TP}{(TP + FN)} \tag{10}$$

False Positive Rate: FPR is defined as the proportion of negative samples (in our case the remaining five events excluding the target event) that are incorrectly identified as positive by the classifier.

$$FPR = \frac{FP}{(FP + TN)} \tag{11}$$

where

TP the number of positive samples that are correctly classified
FP the number of positive samples that are incorrectly classified
FN the number of negative samples that are incorrectly classified
TN the number of negative samples that are correctly classified.

ROC curve has advantage over other techniques in the sense that it visualizes all possible threshold values unlike other metrics such as precision and recall, misclassification rate which consider only a single threshold value. More is the area under the ROC curve, better is the performance of the classifier model.

4 Results and Discussion

The considered model detects the electric signals that occur in the brain just before undertaking a particular hand movement. The convolutional neural network estimates the probability of occurrence of each of the six events for the entire time window of 150 ms associated with the hand movement. Among the predicted values, the event with the maximum probability is chosen as the occurred event by CNN. The final AUROC obtained using the proposed technique is 0.91 which is calculated by taking mean of all the individual ROCs of each of the six events as shown in Fig. 6. The subject-wise AUC values for the each of the 12 subjects is also generated to compare the performance of the classifier among the different subjects shown in Fig. 7.

The detection performance of different classifiers and the proposed model are summarized in Table 2.

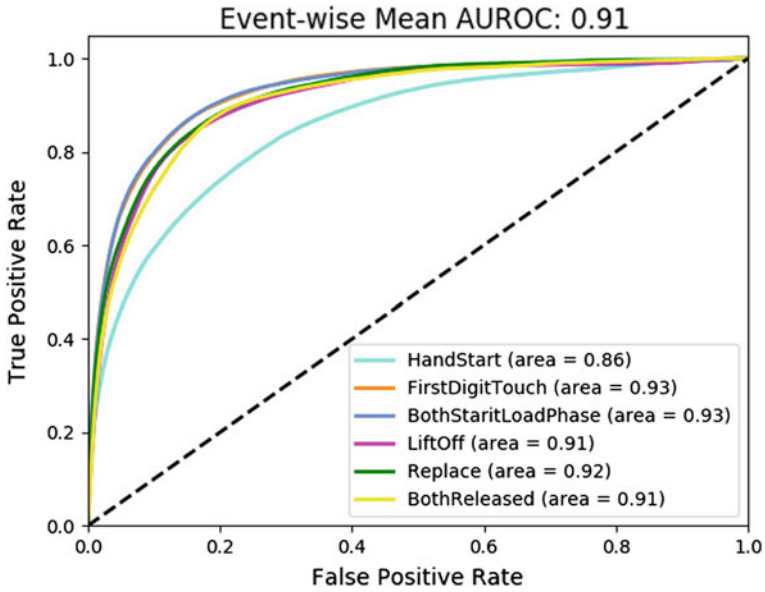


Fig. 6 Event-wise AUROC results

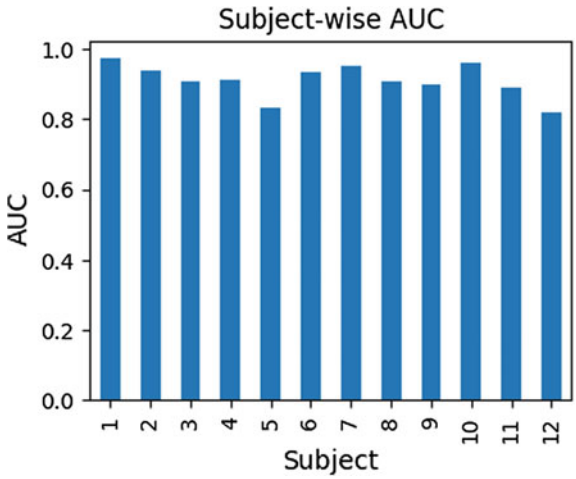


Fig. 7 Subject-wise AUROC result

Table 2 Comparison of different classification algorithms

Algorithm	AUROC std. dev.	AUROC mean
CNN with normalization	0.0475	0.829
Band-pass filter bank CSP with logistic regression	0.053	0.722
Low-pass filter bank with logistic regression	0.0475	0.796
CNN with Butterworth low-pass filtering	**0.002**	**0.91**

5 Conclusion

The proposed technique succeeded in classifying the EEG signals into the six different stages of a grasp-and-lift task with a high AUROC value of 0.91. This is the best result obtained by a single convolutional neural network with minimal signal preprocessing as per the authors' knowledge. Most of the existing methods comprise of a pipeline of numerous neural networks and classifiers along with various signal processing techniques involved. This also makes CNNs an extremely promising choice for the classification and identification of EEG signals. The presented method will increase the detection accuracy of BCI prosthetic hand devices. It would eventually help in improvement of the life of disabled people by providing them the assistance and support for their day-to-day activities.

6 Future Scope

The performance of the model can be further improvised by incorporating band-pass filtering to the signals. Training of the CNN can be performed for more epochs, and learning parameters rho and epsilon can be adjusted to enhance the learning. The convolutional neural network can be made wider or deeper for better results. Dropout layers can be added to control any overfitting if it occurs.

References

1. Butterworth filter (2015). https://www.electronicshub.org/butterworth-filter/
2. Abdulkader, S.N., Atia, A.: Brain computer interfacing: applications and challenges. Egypt. Inf. J. **16**, 213–230 (2015)
3. An, J., Cho, S.: Hand motion identification of grasp-and-lift task from electroencephalography recordings using recurrent neural networks. In: International Conference on Big Data and Smart Computing (BigComp), pp. 427–429. IEEE, Hong Kong (2016)

4. Huang, J., Ling, X.C.: Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. Knowl. Data Eng. **17**(3), 299–310 (2005)
5. Klem, G.H., Luders, H.O., Jasper, H., Elger., C.: The ten-twenty electrode system of the international federation. Int. Fed. Clin. Neurophys. (Elsevier Science B.V.) **10**, 371–375 (1999)
6. Liao, K., Xiao, R., Gonzalez, J., Ding, L.: Decoding individual finger movements from one hand using human EEG signals. PLOS ONE **9**, 1–12 (2014)
7. Lotte, F., Congedo, M., Lecuyer, A., Lamarche, F., Arnaldi, B.: A review of classification algorithms for EEG-based brain computer interfaces. J. Neural Eng. **4**(2), 1–24 (2007)
8. Luciw, M.D., Jarocka, E., Edin, B.B.: Multi-channel EEG recordings during 3,936 grasp and lift trials with varying weight and friction (2014). https://www.nature.com/articles/sdata201447
9. Nicolas-Alonso, L.F., Gomez-Gil, J.: Brain computer interfaces, a review. Sensors **12**(2), 1211–1279 (2012)
10. Tabar, Y.R., Halic, U.: A novel deep learning approach for classification of EEG motor imagery signals. J. Neural Eng. **14**(1) (2016)
11. Varszegi, K.: Comparison of algorithms for detecting hand movement from EEG signals. In: International Conference on Systems, Man, and Cybernetics, pp. 2208–2213. IEEE, Budapest, Hungary (2016)
12. Wolpaw, J.R., Birbaumer, N., Heetderks, W.J.: Brain computer interface technology: a review of the first international meeting. IEEE Trans. Rehabil. Eng. **8**(2), 164–173 (2000)
13. Zeiler, M.D.: Adadelta: an adaptive learning rate method. Comput. Res. Repos. (CoRR) (2012). arXiv:1212.5701 [cs.LG]
14. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision 2014. LNCS, vol. 8689, pp. 818–883 (2014)

Neural Net Optimization by Weight-Entropy Monitoring

Seba Susan, Rohit Ranjan, Udyant Taluja, Shivang Rai and Pranav Agarwal

Abstract A novel technique of monitoring the entropy of network weights is proposed for optimizing the multilayer perceptron neural network classifier. The set of weights associated with the input of every perceptron are normalized to a probability distribution, and the entropy of weights is computed for the whole network using chain rule. The synaptic weights being initially random start converging to definite values in epoch training. The stopping criterion for the gradient-based backpropagation (BP) optimization algorithm is defined by the stabilization of the entropy of weights over a time window, even though the cost function continues to steadily decline. In the case of neural networks trained by the gradient-free particle swarm optimization (PSO), the point of convergence is interpreted as the particle position at which the entropy of weights is minimum that corresponds to the most uneven network weight distribution. The entropy used in our experiments is the non-extensive entropy with Gaussian gain that is nonadditive when used in a summation. Experimental results on benchmark datasets from the UCI repository indicate a quicker convergence of the optimization process in both instances, with the high accuracies of classification maintained.

Keywords Neural network · Multilayer perceptron · Backpropagation
Particle swarm optimization · Non-extensive entropy with Gaussian gain

S. Susan (✉) · R. Ranjan · U. Taluja · S. Rai · P. Agarwal
Department of Information Technology, Delhi Technological University,
Bawana Road, New Delhi 110042, India
e-mail: seba_406@yahoo.in

R. Ranjan
e-mail: rohitr1612@gmail.com

U. Taluja
e-mail: talujaudyant@gmail.com

S. Rai
e-mail: shivangrai95@gmail.com

P. Agarwal
e-mail: pranavdtu061@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_16

1 Introduction

The supervised learning in the multilayer perceptron (MLP) feedforward neural network can be viewed as an optimization problem for determining the optimal synaptic weights that minimizes the cost function in the shortest time possible [1]. The cost function is interpreted as the error between the target or desired output and the actual network output. The determination of optimal weights for a neural net finds wide applications in prediction and forecasting [2], diagnosis of medical ailments [3], stock market forecasting [4], automatic speech recognition [5], satellite image processing [6], and Ethernet traffic prediction [7]. The optimality of the synaptic weights determined in the training phase is evaluated in the testing phase, with a trade-off required between the convergence time in the training phase and the accuracies obtained in the testing phase [8]. Conventional learning of the weights of the multilayer perceptron involves gradient-based optimization methods such as the backpropagation (BP) [9] which is based on the steepest descent algorithm [10]. The problems associated with gradient-based methods are slow convergence [11] and getting trapped in local minima [12]. Evolutionary computing-based optimization techniques that seek to converge to a globally minimum solution are a good substitution to the BP algorithm, for example particle swarm optimization [13, 14], hybrid particle swarm optimization and gravitational search [15], genetic algorithm [16] and simulated annealing [17]. The optimization process is terminated if the cost function falls below a user-specified threshold that is usually empirically determined and is constant for all datasets. Alternative stopping criteria for quick convergence are proposed in the works of [18] by using cross-validation [19] that proposes acceleration through weight extrapolating and [20] that advocates sensitivity analysis for fast learning. In this work, we propose a novel stopping criterion for neural network optimization algorithms, based on the weighted sum of entropies of network weights plotted as a function of time during the training phase. For this purpose, we use the non-extensive entropy with Gaussian gain [21] that is nonadditive when used in a summation. The paper is organized as follows: The computation of the entropy of the weights of a multilayer perceptron network using the chain rule of entropy is described in Sect. 2. The proposed stopping criteria for derivative-based and derivative-free network optimization algorithms based on the entropy of network weights are described in Sect. 3. The experimental results are analyzed in Sect. 4, and the overall conclusions are drawn in Sect. 5.

2 Computation of the Entropy of Network Weights Using the Chain Rule of Entropy

The degree of randomness of the outcomes of a probabilistic experiment with n-possible outcomes is measured by the entropy measure. Non-extensive entropy measures have been used before for event and anomaly detection [22–26], primarily

due to their nonadditive behavior. When the non-extensive entropies of component sub-systems are added up, and this summation is monitored over time, a significant deviation or spike in the graph is observed only for a significant change or anomaly in the input [22]. We focus on the one entropy that has been adapted successfully in the past for such applications [22–25]—the non-extensive entropy with Gaussian gain [21] that is defined as the weighted average of Gaussian information gains of a probability distribution $P = \{p_k\}$, given by

$$H(P) = \sum_{k=1}^{n} p_k e^{-p_k^2} \tag{1}$$

By itself, it is a good regularity indicator and can identify different types of repetitive patterns occurring in natural textures [21]. Its highlight is its nonadditive behavior when used in a summation. Being exponential, the entropy in (1) is non-additive which means that when entropies of two different sub-systems A and B are added together the joint entropy is always lesser than the sum of individual entropies, i.e.,

$$H(A, B) < H(A) + H(B) \tag{2}$$

This implies that the sum contributions of all sources are not additive and only those contributions that are significant enough will make an impact in the summation, while those contributions which are insignificant and could be interpreted as noise or false alarms are subdued or ignored. This feature of the entropy in (1) is used for monitoring crowd motion in [22], the occurrence of anomalic training samples in a dynamic neural network [23], for determining the optimum window size enclosing texture defects in [24] and for locating anomalic facial intensities related to emotions in [25]. In our present application, the summation of non-extensive entropies is encountered when we apply the chain rule of entropy to the MLP neural network structure. The chain rule-based entropy computation in the MLP neural network is a part of the work done in [23], and this serves as a precursor to the entropy computation within our neural network structure. The chain rule of entropy is next explained below in detail.

2.1 Chain Rule of Entropy

The chain rule of entropy was proposed by Shannon [27] to find the overall entropy of a system comprising of sub-systems forming a chain of conditional dependencies. Consider the finite state machine shown in Fig. 1, with the transition probabilities shown along the arrows.

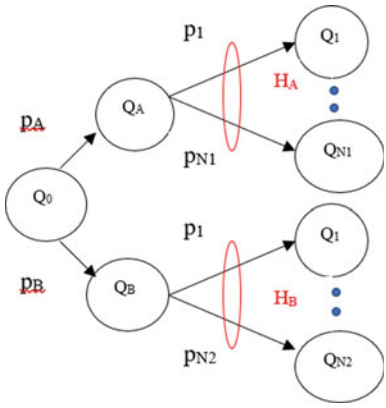


Fig. 1 A finite state machine structure for the applying the chain rule of entropy

Then as per the chain rule of entropy, the overall entropy of the source is given by

$$H(S) = p_A H(A) + p_B H(B) \tag{3}$$

Substituting for the entropies in (3) with the non-extensive entropy in (1), we have

$$H(S) = p_A \sum_{a=1}^{N_1} p_a e^{-p_a^2} + p_B \sum_{b=1}^{N_2} p_b e^{-p_b^2} \tag{4}$$

The entropy computation in (4) is valid if the three probability distributions involved are complete; i.e., the sum of the probabilities in each of these probability distributions sums to one: $(p_A + p_B = 1,\ \sum_{a=1}^{N_1} p_a = 1,\ \sum_{b=1}^{N_2} p_b = 1)$.

2.2 *The Chain Rule of Entropy Applied to the MLP Neural Net Structure*

Now, consider the computation of the chain rule of entropy in the multilayer perceptron neural network architecture, as proposed in [23]. Consider a single neuron (namely the jth neuron) shown in Fig. 2 that belongs to the hidden layer and the associated weights $\boldsymbol{w_{kj}} = \{w_{kj}\}$ of the n lines input to the jth neuron where $k = 1, 2, \ldots, n$.

In order to measure the randomness of the weights associated with the jth neuron using entropy, the weights are converted to a probability distribution by normalizing them in the range of 0–1 and ensuring that their sum equals to 1 as shown in (5) and (6).

$$\overline{w_{kj}} = \frac{w_{kj} - \min(\boldsymbol{w_{kj}})}{\max(\boldsymbol{w_{kj}}) - \min(\boldsymbol{w_{kj}})} \tag{5}$$

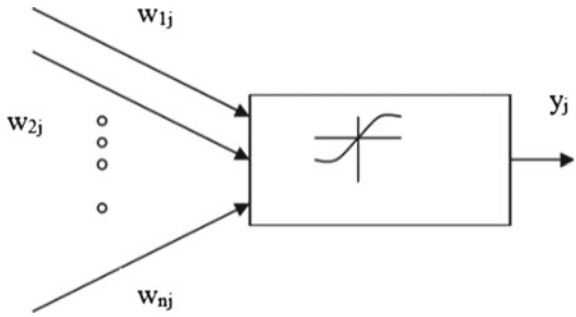


Fig. 2 jth neuron and the weights associated with its input lines

$$p_k^{(j)} = \frac{\overline{w_{kj}}}{\sum_{i=1}^{n} \overline{w_{ij}}} \tag{6}$$

where $k = 1, 2, \ldots, n$ and $\sum_{k=1}^{n} p_k^{(j)} = 1$.

The entropy of the network weights associated with the jth neuron is obtained by substituting the probability values $\{p_k^{(j)}\}$ in (1) as shown below.

$$H_j = \sum_{k=1}^{n} p_k^{(j)} e^{-\left(p_k^{(j)}\right)^2} \tag{7}$$

The entropy of the overall n-H-R multilayer perceptron network having an n-dimensional input, H hidden neurons in the hidden layer, and R outputs/output neurons is computed using the chain rule of entropy as shown below.

$$H_{Net} = \sum_{r=1}^{R} \sum_{j=1}^{H} p_j^{(r)} H_j \tag{8}$$

Here, $p_j^{(r)}$ is the probability associated with the jth input line of the rth output neuron. The nonadditivity property holds for the summation in (8) due to the nonlinear exponential term in (7), making it a useful measure for continuous monitoring in network optimization algorithms where weights evolve and converge over time.

3 Proposed Optimization of Network Weights by Weight-Entropy Monitoring

In this section, we present the methodology for the proposed weight-entropy monitoring scheme in feedforward neural networks. In batch or epoch training, the weight adjustment is performed after the entire training set is presented. The cost function is

averaged over the entire training set. For the derivative-based optimization process, the weight vector traces a well-defined trajectory in the weight space. Convergence in this case implies that the expected value of the error is lower than a threshold value that is preset by the user or if the maximum number of iterations is reached [1]. The network weights start converging to definite values as the training progresses and the error gets continually reduced. Hence for the backpropagation (BP) algorithm [9], we propose to test for constancy in the weighted sum of entropies in (8) as the stopping criterion for the optimization process. The entropies should be constant for a prescribed number of successive samples (a choice of seven samples works for our experiments) in the time window as the training progresses.

On the other hand, for the derivative-free optimization methods such as particle swarm optimization (PSO) [28], the weight vector traces a random trajectory in the weight space since multiple neural networks are at work, each corresponding to a particle, and the best solution is sought which may be quite different from that of the previous iteration [29]. The convergence of this algorithm is much faster than its derivative-based counterpart [30]. So, in order to seek alternative stopping criteria other than the cost function that constantly declines with each iteration, we rely on the entropy computed from each neural network/particle using (8). The lower this entropy, the more uneven are the input weights associated with each neuron which in turn implies a better feature ranking and selection since it assigns lower weightage to those input lines that do not contribute much to the output [31]. An uneven weight structure when achieved points to a good understanding of the input features which naturally translates to higher classification accuracies. The sum of the entropies of all the particles/neural networks will indicate a negative spike at this point corresponding to the global minima solution.

We describe below the methodology for incorporating weight-entropy monitoring in neural networks that are trained by the BP and PSO algorithms.

3.1 Steps for Incorporating Weight-Entropy Monitoring in Feedforward Backpropagation Neural Networks

- Find the output of the feedforward neural network after one complete epoch of training samples has been presented, and compute the average cost function.
- Update the synaptic weights based on the gradient of the average cost function.
- Convert the input weights associated with each neuron into probability distribution, and compute the entropy associated with each neuron as per Eqs. (5)–(7).
- Find the weighted sum of non-extensive entropies emanating from all the output nodes as per Eq. (8).
- Plot the weighted sum of entropies versus epoch as training progresses.
- Continually check whether the entropy values stabilize (standard deviation (s.d.) of entropies = 0) as the epoch training progresses for every overlapping consecutive *seven* samples along the time axis.

- At the very first point where the standard deviation of entropies equals to zero, stop the algorithm, and the network weights are considered optimized at this point.

3.2 Steps for Incorporating Weight-Entropy Monitoring in Artificial Neural Networks Trained by Particle Swarm Optimization (ANN-PSO)

- For each particle, initialize a feedforward neural network with the weights being the position vector of the particle.
- Find the output of each feedforward neural network after one complete batch of training samples has been presented, and compute the average cost function of each network. Find the local best position of each particle and the overall global best based on the fitness function values.
- Update the position vector or synaptic weights of each particle based on the local and global best positions as per the procedure outlined in [28].
- Convert the input weights associated with each neuron into probability distribution, and compute the entropy associated with each neuron as per Eqs. (5)–(7).
- Find the weighted sum of non-extensive entropies emanating from all the output nodes as per Eq. (8).
- Find the sum of entropies emanating from all particles/networks.
- Plot the weighted sum of entropies of all particles versus epoch as training progresses.
- Find the first local maximum in the entropy graph. This is required since the weights during some initial iterations show an initial drop in entropy and then the entropy rises to a local peak as the weights are randomized and then finally as the particles start moving toward the local and global best solutions, the entropy of weights drops again.
- Find the local minimum that occurs after the first local maximum, and the algorithm could be possibly terminated here. Alternatively, the algorithm could continue and other local minima could be considered as the stopping point if their entropy value is lower than that of the previously recorded local minimum. The point of minimal entropy in the overall temporal graph indicates the most optimal solution of all, and on attaining this valley point the algorithm is considered to be converged and the weights are considered to be optimal.

4 Experimental Results and Discussions

The experiments are performed in C++ software on a 2.6 GHz PC. We use five datasets: *Iris*, *Wine*, *Ionosphere, Breast Cancer, Seeds* datasets from the UCI repository [32]. Alternate samples are considered for training and testing. Two different

weight optimization processes, backpropagation and particle swarm optimization, are used for the evaluation of our weight-entropy monitoring scheme. Two different cost functions—mean square error (MSE) and cross-entropy error (CCE)—are tested. Our multilayer perceptron (MLP) architecture contains a single hidden layer with seven hidden neurons.

The entropy monitoring results for feedforward backpropagation for the *Iris* dataset are shown in the form of graphs in Fig. 3 for cross-entropy error (CCE) as the stopping criterion. The CCE graph in Fig. 3a shows a constant decline. The classification accuracy chart in Fig. 3d on the other hand shows an early convergence which goes undetected in conventional learning. The proposed entropy of the weights computed by (8) plotted over time in Fig. 3b shows an initial high followed by a drop in the entropy values which later rises and stabilizes to a constant value. A comparison of the accuracy graph in Fig. 3d and the entropy graph in Fig. 3b shows a remarkable coincidence between the stabilization of the entropy of weights and that of the accuracies at around 3688 iterations, though the actual CCE cost function in Fig. 3a constantly declines. This confirms the veracity of our claim for weight-entropy monitoring being a reliable means for early termination of the optimization process. As per the procedure outlined in the previous section, the point of convergence is determined from a constancy of entropies (s.d. = 0) over a seven-sample temporal window as shown in Fig. 3c.

In case of supervised learning by particle swarm optimization (PSO), the minimum entropy point in the temporal graph indicates the optimal result. Figure 4 demonstrates the results of weight-entropy-based monitoring of the artificial neural network optimized by particle swarm optimization (ANN-PSO) for the *Breast Cancer* dataset. The correlation between the lowest point in the entropy graph and the optimality of the results of the neural network at this point is demonstrated by these graphs.

The details of training the feedforward neural networks and subsequent classification accuracies are summarized in Table 1 for the various datasets and optimization methods. The maximum number of iterations is set to 50,000 for BP and 7000 for PSO. The error threshold for stopping the conventional BP and PSO algorithms is preset by the user as seen in Table 1 and varies for different datasets, based on empirical hits and trials. For backpropagation optimization, the proposed algorithm converges faster due to the stabilization of entropy values. The stopping point for our algorithm is computed automatically and is not user-defined for different datasets. The reduction in the number of epochs from 27,260 to 4575 for backpropagation for *Iris* is an example. In the case of network optimization by PSO, the number of epochs is already less and the proposed entropy-based stopping criterion makes the artificial neural network with particle swarm optimization (ANN-PSO) even faster without any loss of accuracy, as evident from the results for the datasets in Table 1.

For ANN-PSO, the number of initial particles varies from 10 to 27 for different datasets in the conventional case, while the number of hidden neurons is set to 7. The entropy analysis irrespective of the experimental setting leads to early termination of the optimization process with the high accuracies maintained. After attaining the minimum point after the first local peak or maximum, the temporal graph could be

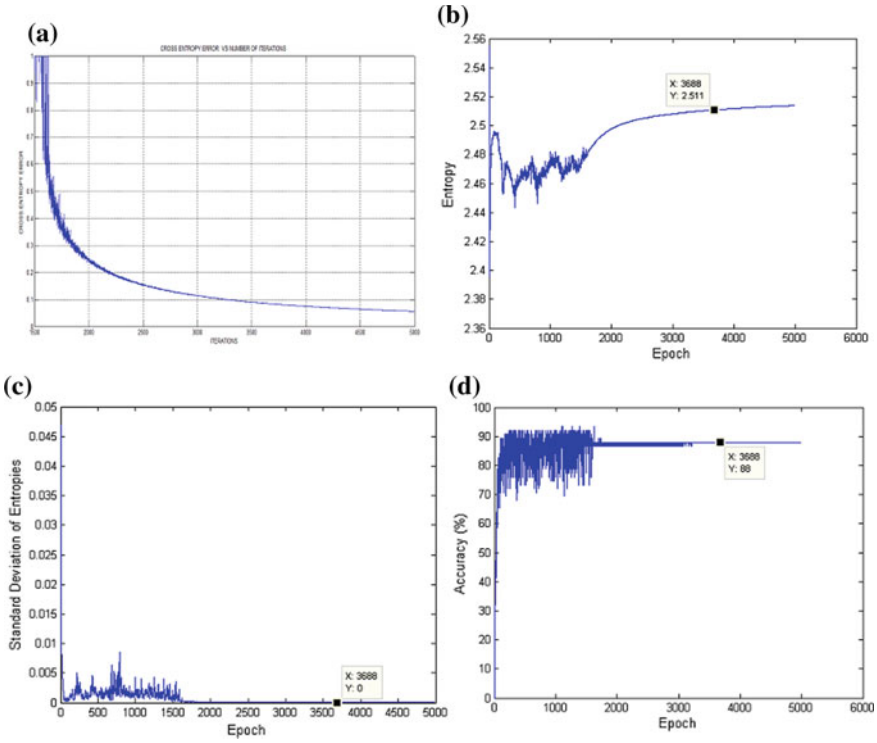


Fig. 3 Weight-entropy monitoring graphs for *Iris* dataset for backpropagation using cross-entropy error as the stopping criterion, **a** cross-entropy error (CCE) graph, **b** entropy versus epoch, **c** standard deviation of entropies over a seven-sample temporal window, **d** accuracy versus epoch

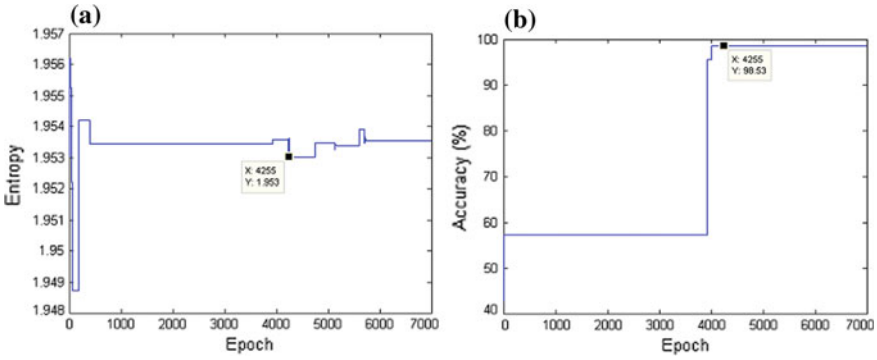


Fig. 4 Weight-entropy monitoring graphs for the *Breast Cancer* dataset for ANN-PSO using cross-entropy error as stopping criterion, **a** entropy versus epoch and **b** accuracy versus epoch

Table 1 Classification results (in %) for various datasets of UCI repository [32] for various feed-forward neural network configurations and optimization methods

Dataset	Optimization method	Stopping criterion (user-defined threshold for conventional optimization algorithms)	Number of epochs	Accuracy (%)
Iris	Backpropagation	Mean square error (0.004)	27,260	85.33
Iris	Backpropagation	Mean square error with entropy monitoring (proposed)	4575	85.33
Iris	Backpropagation	Cross-entropy error (0.004)	49,473	88
Iris	Backpropagation	Cross-entropy error with entropy monitoring (proposed)	3688	88
Iris	Particle swarm optimization	Mean square error (0.099)	105	98.67
Iris	Particle swarm optimization	Mean square error with entropy monitoring (proposed)	105	98.67
Iris	Particle swarm optimization	Cross-entropy error (0.099)	111	98.67
Iris	Particle swarm optimization	Cross-entropy error with entropy monitoring (proposed)	108	98.67
Ionosphere	Particle swarm optimization	Mean square error (0.04)	23	97.78
Ionosphere	Particle swarm optimization	Mean square error with entropy monitoring (proposed)	15	97.78
Ionosphere	Particle swarm optimization	Cross-entropy error (0.04)	118	97.78
Ionosphere	Particle swarm optimization	Cross-entropy error with entropy monitoring (proposed)	101	95.55
Wine	Particle swarm optimization	Cross-entropy error (0.04)	7000 *(max)*	64.04
Wine	Particle swarm optimization	Cross-entropy error with entropy monitoring (proposed)	100	64.04
Breast Cancer	Particle swarm optimization	Cross-entropy error (0.04)	7000 *(max)*	98.53
Breast Cancer	Particle swarm optimization	Cross-entropy error with entropy monitoring (proposed)	4255	98.53
Seeds	Particle swarm optimization	Cross-entropy error (0.25)	444	92.5
Seeds	Particle swarm optimization	Cross-entropy error with entropy monitoring (proposed)	104	90

monitored continually for checking for a further dip in entropy values. In such a case, the minimum position is updated and the search continues, else the last local minimum point is used as the optimal solution point for the weight vector. It is interesting to note from Fig. 4 that the local minima in the entropy graph mostly correspond to local maxima in the accuracy graph. The entropy monitoring of ANN-PSO could be thus terminated at any point with the last minimum point being the most optimal solution. In our graphical results in Fig. 4, and for those in Table 1, we have shown continuous entropy monitoring without termination for ANN-PSO to showcase the maximum accuracies that could be possibly obtained. We observed that the maximum accuracy is obtained at the position of the global minimum of entropy values. The advantage of the proposed entropy monitoring is particularly noticeable for the cases of *Wine* and *Breast Cancer* in Table 1 in which the conventional ANN-PSO does not converge at all due to constancy in error values, and the maximum iteration count is reached. The only cases where the accuracies fall slightly below the maximum are in the case of *Ionosphere* and *Seeds* datasets for ANN-PSO. In all other cases, the highest accuracies are maintained at the early convergence point.

5 Conclusion

A novel stopping criterion for the optimization process in MLP neural networks is proposed that is based on monitoring the entropy of network weights as training progresses. The weighted sum of non-extensive entropies emanating from the different nodes in the network is computed using the chain rule of entropy. For the gradient-based backpropagation, the point of convergence of the weights is determined by the stabilization of the weight-entropy graph over a seven-sample temporal window. In case of evolutionary computing-based particle swarm optimization, the minimum weight-entropy in the temporal graph indicates the most optimal weight distribution. The proposed schema incorporating weight-entropy monitoring gives accurate results and a faster convergence of weights in much lesser number of epochs as compared to the respective baseline methods.

References

1. Haykin, S.: Neural networks: a comprehensive foundation. In: Neural Networks, 2nd ed. Prentice Hall (2004)
2. Maier, H.R., Dandy, G.C.: Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environ. Model Softw. **15**(1), 101–124 (2000)
3. Amato, F., López, A., María Peña-Méndez, E., Vaňhara, P., Hampl, A., Havel, J.: Artificial neural networks in medical diagnosis. J. Appl. Biomed. **11**(2), 47–58 (2013)
4. Ticknor, J.L.: A Bayesian regularized artificial neural network for stock market forecasting. Expert Syst. Appl. **40**(14), 5501–5506 (2013)

5. Deng, L., Hinton, G., Kingsbury, B.: New types of deep neural network learning for speech recognition and related applications: an overview. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8599–8603. IEEE (2013)
6. Linares-Rodriguez, A., Ruiz-Arias, J.A., Pozo-Vazquez, D., Tovar-Pescador, J.: An artificial neural network ensemble model for estimating global solar radiation from Meteosat satellite images. Energy **61**, 636–645 (2013)
7. Park, D.-C.: Structure optimization of bilinear recurrent neural networks and its application to ethernet network traffic prediction. Inf. Sci. **237**, 18–28 (2013)
8. Leboucher, C., Siarry, P., Le Ménec, S., Shin, H.-S., Chelouah, R., Tsourdos, A.: An enhanced particle swarm optimisation algorithm combined with neural networks to decrease computational time. In: Swarm Intelligence Based Optimization, pp. 139–156. Springer International Publishing (2014)
9. Rumelhart, D.E., McClelland, J.L., PDP Research Group: Parallel Distributed Processing, vol. 1. MIT Press, Cambridge, MA, USA (1987)
10. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, vol. 87. Springer Science & Business Media, 2013
11. Zhang, J.-R., Zhang, J., Lok, T.-M., Lyu, M.R.: A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Appl. Math. Comput. **185**(2), 1026–1037 (2007)
12. Gori, M., Tesi, A.: On the problem of local minima in backpropagation. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 76–86 (1992)
13. Van den Bergli, F., Engelbrecht, A.P.: Cooperative learning in neural network using particle swarm optimization. South Afr. Comput. J. **26**, 84–90 (2000)
14. Cells, M., Rylander, B.: Neural network learning using particle swarm optimization. Adv. Inf. Sci. Soft Comput. 224–226 (2002)
15. Mirjalili, S.A., Mohd Hashim, S.Z., Sardroudi, H.M.: Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. Appl. Math. Comput. **218**(22), 11125–11137 (2012)
16. Montana, D.J., Davis, L.: Training feedforward neural networks using genetic algorithms. IJCAI **89**, 762–767 (1989)
17. Shaw, D., Kinsner, W.: Chaotic simulated annealing in multilayer feedforward networks. In: 1996 Canadian Conference on Electrical and Computer Engineering, vol. 1, pp. 265–269. IEEE (1996)
18. Prechelt, L.: Automatic early stopping using cross validation: quantifying the criteria. Neural Netw. **11**(4), 761–767 (1998)
19. Kamarthi, S.V., Pittner, S.: Accelerating neural network training using weight extrapolations. Neural Netw. **12**(9), 1285–1299 (1999)
20. Castillo, E., Guijarro-Berdiñas, B., Fontenla-Romero, O., Alonso-Betanzos, A.: A very fast learning method for neural networks based on sensitivity analysis. J. Mach. Learn. Res. **7**, 1159–1182 (2006)
21. Susan, S., Hanmandlu, M.: A non-extensive entropy feature and its application to texture classification. Neurocomputing **120**, 214–225 (2013)
22. Susan, S., Hanmandlu, M.: Unsupervised detection of nonlinearity in motion using weighted average of non-extensive entropies. SIViP **9**(3), 511–525 (2015)
23. Susan, S., Dwivedi, M.: Dynamic growth of hidden-layer neurons using the non-extensive entropy. In: 2014 Fourth International Conference on Communication Systems and Network Technologies (CSNT), pp. 491–495. IEEE (2014)
24. Susan, S., Sharma, M.: Automatic texture defect detection using Gaussian mixture entropy modeling. Neurocomputing **239**, 232–237 (2017)
25. Susan, S., Aggarwal, N., Chand, S., Gupta, A.: Image coding based on maximum entropy partitioning for identifying improbable intensities related to facial expressions. Sādhanā **41**(12), 1393–1406 (2016)
26. Yin, M., Yao, D., Luo, J., Liu, X., Ma, J.: Network backbone anomaly detection using double random forests based on non-extensive entropy feature extraction. In: 2013 Ninth International Conference on Natural Computation (ICNC), pp. 80–84. IEEE (2013)

27. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
28. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
29. https://visualstudiomagazine.com/articles/2013/12/01/neural-network-training-using-particle-swarm-optimization.aspx. Accessed 11 Nov 2017
30. Gudise, V.G., Venayagamoorthy, G.K.: Comparison of particle swarm optimization and back-propagation as training algorithms for neural networks. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS'03. IEEE (2003)
31. DelSole, T., Yang, X., Tippett, M.K.: Is unequal weighting significantly better than equal weighting for multi-model forecasting? Q. J. R. Meteorol. Soc. **139**(670), 176–183 (2013)
32. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2010). http://archive.ics.uci.edu/ml

Multimodal Data Fusion and Prediction of Emotional Dimensions Using Deep Neural Network

Dhruv Bhandari, Sandeep Paul and Apurva Narayan

Abstract This paper proposes a novel deep neural network model to handle multimodal data. The proposed model seamlessly facilitates fusion of multimodal inputs and provides dimensional reduction of the input feature space. The architecture employs modified stacked autoencoder in conjunction with multilayer perceptron-based regression model. Two variants of architecture are proposed, and experiments have been performed on the multimodal benchmark data (RECOLA) to study the impact of multimodality as against a single modality. Experiments are also conducted to illustrate the effect of presenting multimodal data in sequential or concatenated manner. The results obtained are encouraging. The proposed approach is computationally less expensive than the existing approaches, and the performance is better or at par with other techniques.

Keywords Multimodal data · Stacked autoencoder · Deep neural networks Data fusion

1 Introduction

Humans interact with the environment using multiple senses. Most common among them are auditory and visual senses which often work in conjunction with each other. Reception of audio-visual input forms a perception of occurrence of an isolated event. Perception in this context is an individual conscious experience that combines input from all relevant senses and prior knowledge. The information is integrated and

D. Bhandari · S. Paul (✉)
Dayalbagh Educational Institute, Dayalbagh, Agra, India
e-mail: spaul@dei.ac.in

D. Bhandari
e-mail: dei.dhruvbhandari@gmail.com

A. Narayan
University of Waterloo, Waterloo, Canada
e-mail: apurva.narayan@uwaterloo.ca

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_17

treated as a unitary representation of the world. Adding together response obtained from each modality at a time would still not be equal to the response obtained by multimodal stimulus. The cross-modal integration of information is more accurate than information presented by single modality [1]. Thus, this super-additive effect of multisensory integration needs to be addressed using a better model.

The advancements in machine learning techniques are gradually making computers intelligent. However, they are still not very efficient in perceptual tasks which humans can perform. The recent success of deep neural networks in solving problems on computer vision [2] and audio independently, motivates us to work on the idea of using them for the above-explained multimodal fusion problems. The focus has now shifted toward building models which aim to simulate human perception and use cross-modalities to understand the environment in an efficient way [3–7]. One such framework which uses multimodal deep neural network for audio-visual emotion recognition has been presented in [3]. Many hybrid models are also gaining huge success in this field of multimodal data fusion. A comparison of the results between a hybrid model approach and traditional deep neural network models has been shown in [4]. An interesting application of multimodal data fusion is the analysis of depression and emotion recognition. Depression needs immediate medical supervision. Even small stimuli can cause a cascading effect when it comes to change in actions or behavior for a patient in emotionally unstable states. Previous studies show that dimensional emotions (arousal and valence) are more suitable to express subtle but complex emotional states as compared to discrete emotions (happy, sad, angry, etc.) [8–11]. Arousal and valence are two such dimensional emotions. Arousal on the one hand describes how excited or apathetic any emotion is, whereas valence refers to how positive or negative an emotion is [12].

This paper focuses on predicting continuous arousal and valence values for multimodal input (audio–video) using deep neural networks. Results of various combinations of input like audio only, video only, and two multimodal audio–video representations have been compared. Multiple fusion techniques have been shown using stacked autoencoders. A simple but effective multilayer perceptron (MLP) model has been used as a regression tool to predict dimensional emotions. The performance of our model has been tested on a benchmark dataset (RECOLA dataset) [13].

The major contributions of this paper lie in development of deep neural network model which seamlessly handles multimodal information, facilitates multimodal data fusion, and performs dimensional reduction for regression and classification problems. The proposed approach is computationally less expensive (as explained in Sects. 2 and 3) than the existing approaches, and the performance is better or at par with the other techniques. At the implementation level, a customized layer is built in the frameworks used—Keras and TensorFlow [14, 15]. This is a hybrid layer which acts intelligently according to the input it receives. Its functionality becomes prominent dealing with multimodal data in the input in Sects. 3 and 4. The remaining paper is organized as follows: Sect. 2 provides the required theoretical detail of proposed approach. In Sect. 3, the dataset used, experimental setup and the conducted experiments have been explained. Section 4 presents the results from experiments. Finally, Sect. 5 concludes the paper with discussion and the future work.

2 Proposed Approach and Methodology

This section introduces the proposed approach and the platforms that have been used to develop the models. The required theoretical details have been explained, and the particulars of implementation on the platforms—Keras and TensorFlow—have been discussed.

Autoencoders are used for learning the hidden structure present in the input data. They have also been used as effective retrieval models [16]. They try to reconstruct the input at the output while satisfying some constraints. The network tries to learn an identity function such that:

$$Y_{(w,b)}(x) \cong x. \tag{1}$$

where 'Y' is the output of the autoencoder, 'x' is the input, and 'w' and 'b' are the weights and the bias of the network. The identity function seems to be trivial to learn, but in the presence of some constraints on the network it can be used to discover interesting structures in the input data. When the input data is completely random, learning the identity function is difficult. But when the input features have some correlation between them, with sufficient training, it is possible to learn this function and obtain a compressed representation of the input data.

2.1 Modified Stacked Autoencoder

A modified version of stacked autoencoders (SAE) is presented for integrating and synthesizing information from multiple modalities specifically audio and video signals. The goal is to bring about an effective multimodal retrieval using the bottleneck constraints of stacked autoencoders (the number of neurons in hidden layer is less than input layer neurons). In cases with multimodal input, it becomes necessary to preserve the intra-modal representations and extensively learn the intermodal representations. SAEs are used to learn the features from both the modalities—audio and video. The model is initiated with individual modalities and is gradually built into a multimodal framework. The input audio features are mapped to a latent audio space using the SAE as shown in Fig. 1. A similar SAE is used to map features from the video input to a latent video space. In this paper, latent space is the dimensional space of the output at layer E2 in Fig. 1. It is called latent because the feature space is not immediately apparent. It is discovered and populated using autoencoder from the feature sets of different modalities as inputs. This also represents a fused feature set. The proposed architectures of SAEs revolve around the fact that information present in a video frame at any time instant will generally be more than that in the audio frame. Thus, the number of neurons in the input in Fig. 1 for audio encoder will be less than that of the video encoder. The dimensionality of the latent space has to be preserved so that the two modalities can be compared easily using the same

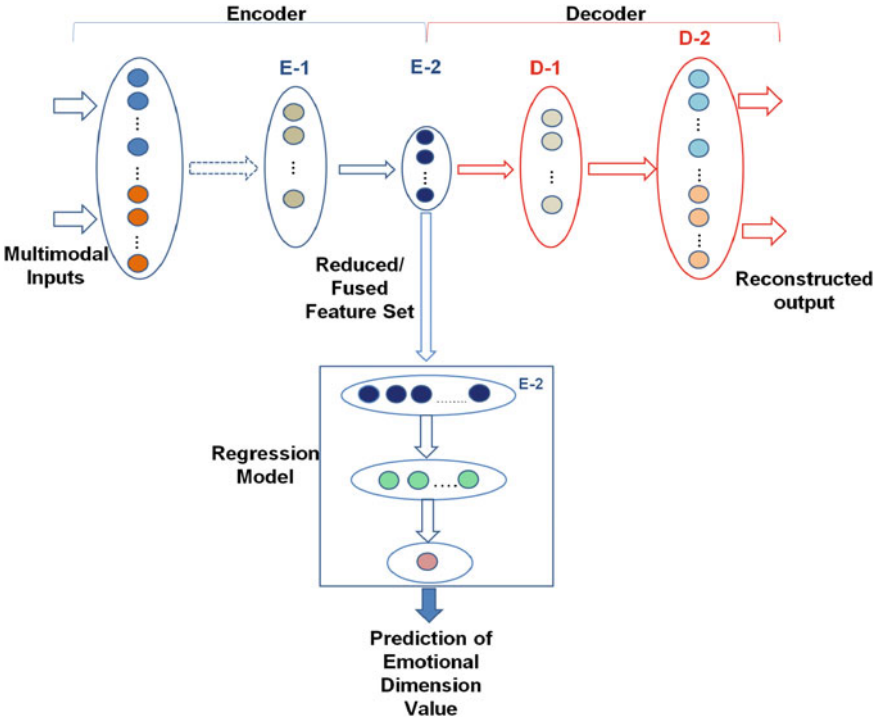


Fig. 1 Stacked autoencoder for fused feature set with regression model

MLP-based regression model. Hence, the number of neurons in layer E2 is kept same for both modalities.

The individual modalities are fairly straightforward to comprehend. But while working with features from multiple modalities, the architecture has to be modified to accommodate information simultaneously (from all of them). The modifications should be such that the same network can be used for both the modalities. The network architecture should share weights between modalities to facilitate a combined learning and a combined representation in the latent space. The focus is on learning the interclass representations without compromising with intra-class representations. As stated earlier, an audio frame will have less information than a video frame, at any given time. Using the same network for both modalities will mean that the audio network is a subset of video network—in terms of parameters (weights, bias). The idea is to nullify a small part of the network corresponding to input dimension which is not being used. The neurons and connections between layers, which represent a specific dimension in the input, are dropped, while the others work as usual. The modification is partially similar to the training phase mentioned in [17]. Srivastava et al. propose the idea of dropping out a few neurons randomly in the training phase. They do it to prevent deep neural networks from overfitting. Our aim is to accommodate multimodal information in the same network. A pre-defined number of neurons

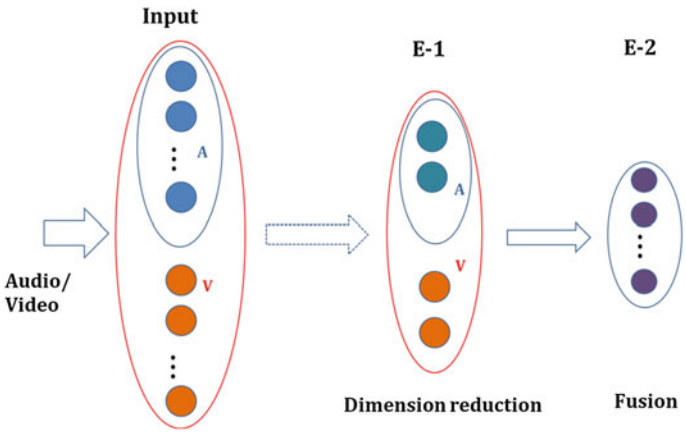


Fig. 2 Type I architecture of autoencoder for multimodal inputs

which represent a specific dimension in the input are dropped. The authors in [17] use the neurons in the test phase by multiplying the output with the pre-defined probability of dropping neurons.

In this paper, the left out neurons are not used in the test phase as well. It implements the idea of a subset network in true sense. The same has been demonstrated in Fig. 2 and is labeled as Type I architecture.

While dealing with the audio modality, the autoencoder section labeled 'A' gets activated and the remaining part is nullified. On the other hand, while working with the video modality, the network 'V' (of which 'A' is a subset) gets activated. It may be noted that the output of the fused feature space is obtained at the output of the layer E2. The complete layer E2 always gets activated as the latent space has the same dimensionality in all experiments in this paper. It should also be noted that the same network is used to train for video as well as for audio modality. Thus, the weights in the network are tuned across both modalities, but sequentially.

The second architecture for multimodal fusion of information is shown in Fig. 3 and is labeled as Type II architecture.

One of the advantages of multimodal fusion is that even if one of the modalities is corrupt to some extent or the information content in any one modality is missing, the other modality can be used to predict the emotional dimension as shown in [5]. The significance of this architecture is explained in Sect. 3.2.

2.2 Keras and TensorFlow

The architectures explained have been implemented in Keras with TensorFlow as a backend [14, 15]. The architecture for audio-only and video-only modalities is implemented using the usual dense layers. A dense layer in Keras is a fully connected

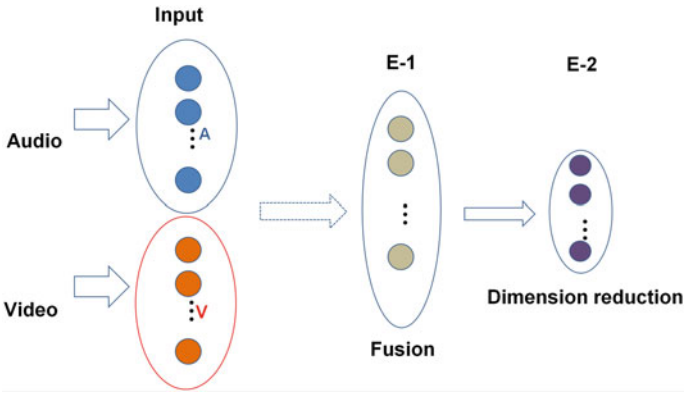


Fig. 3 Type II architecture of autoencoder for multimodal inputs

layer as represented in neural networks. A custom layer was required for Type I and Type II architectures (explained in Sect. 3.2). This layer was designed to handle the requirement of activating only a subset of the network. This gives us the flexibility to use the same network for the multimodal case with augmented inputs. Sharing of weights (in Type I) brings down the total number of parameters that have to be saved during training. It makes the architecture (Type I) memory efficient which otherwise would have consumed almost double the memory.

3 Audio–Video Multimodal Classification

3.1 Dataset: RECOLA

The remote collaborative and affective interactions database has been used for testing our model [13]. The corpus includes the following multimodal signals—audio, video, electrocardiogram (ECG), electrodermal activity (EDA). These are recorded simultaneously from 27 French-speaking subjects of different nationalities—German, French, and Italian. This was done to provide some diversity in the expression of emotion being captured. These 27 subjects were then divided into groups of 9 each—train, development, and test.

The dataset also consists of ground truth values for each of the emotional dimensions—arousal and valence. The labeling of these ground truth values was done by six gender-balanced French-speaking assistants. Values of arousal and valence are recorded using a 40 ms frame. In this paper, the audio features and video appearance features (LGBP-TOP) given in the dataset have been used. A detailed description of these can be found in [8, 13, 18].

3.2 Experiments

The aim of these set of experiments is to show the power of using multiple modalities over single modality in understanding the environment. In this section, the experiments carried out to study fusion of information and hence perception have been explained in detail. The study has been divided into a set of four experiments. First experiments are conducted to find the arousal and valence values for individual modalities (audio and video separately). Then, they are conducted using modified architectures—Type I and Type II for multimodalities. A simple example can be used to understand all the experiments—a group of people are asked to analyze a movie scene by hearing it (video mute), then by watching the video only (audio mute), next by watching the video followed by hearing the audio (audio sequentially after the video), and finally by hearing and watching the scene in sync. The catch here is that the participants do not understand the language being spoken in the scene. They will not be able to comprehend the situation using linguistics. A detailed explanation of the significance of all of these experiments is given below along with the algorithms.

Algorithm 1 : Training Autoencoder (Unsupervised Learning)

1. ***For*** *each layer in stacked autoencoder model***:**
2. **Models** ← ***Pre-Train(Model, Input)***
3. **List_models** ← *Append* **Models** *to List*
4. ***Stack Models*** *from* ***List_Models*** *for* ***Fine Tuning***
5. *Save Stacked Autoencoder Model after fine tuning.*

Algorithm 2 : Training Multi Layer Perceptron (MLP) for Regression (Supervised Learning)

1. ***Load*** *Fine-tuned Stacked Autoencoder Model.*
2. ***For*** *each batch of patterns in Input**:***
3. **Reduced Feature-set** ← *Obtained From Loaded Model.*
4. *Calculate predicted value of emotional Dimension using* **Reduced Feature-set** *as input to MLP.*
5. *Calculate error by comparing with ground truth values given in database for each input.*
6. *Train MLP for Regression using this error through backpropagation.*

Adadelta has been used as the learning technique in all of them [19].

Experiment 1—Using Audio-Only Features as Inputs

This experiment was performed using only the audio features as input to the SAE. The output from the reduced space shown in Fig. 1 goes to an MLP regression model. The arousal and valence values are predicted from the given input. It must be noted that the regression phase is a supervised learning phase. The MLP is trained using the gold standard values provided in the dataset at each instant of time (for arousal and valence separately). This case of experiment is very similar to hearing the movie

scene (video mute). Hence, audio features like modulation in speech, tone of speech can only be used to analyze the scene.

Experiment 2—Using Video-Only Features as Input
This experiment was performed using video-only features as input to the SAE. The procedure is similar to Experiment 1, with the difference being—video features as the only input and the architectural changes associated with it (the number of neurons in input layer is more than that of audio in Experiment 1). An MLP was trained, and continuous real values of arousal and valence were predicted at every time instant. This case is very similar to seeing the video alone (with audio mute). Smiles or anger on the faces can easily be identified after seeing the video.

Experiment 3—Using Sequential Audio and Video Input
In this experiment, Type I architecture (Fig. 2) has been used. The network is first trained for video-only input. Immediately after this, the same network is trained on the audio features. It should be noted that the encoders in Fig. 1 are replaced by the modified encoders shown in Fig. 2. The decoder will just be a mirror image of this new encoder. These form the first modified stacked autoencoder proposed in this paper. The first hidden layer of the modified encoder brings about a dimensional reduction and initiates fusion by sharing weights, while the second hidden layer consolidates the fusion to represent input in the latent space. The rest of the procedure for training an MLP is similar to Experiment 1. This case corresponds to showing only video with audio mute to start with and then making the participants hear the audio track of the video scene. It is again very intuitive that the participants will be able to correlate some of the actions in video with the audio events. This sequential input of two modalities introduces a time lag in the information provided by them. The information from both modalities although in sequential manner is used to predict the final arousal and valence values.

Experiment 4—Using Concatenated Audio and Video Input
This experiment uses Type II architecture (explained in Sect. 2.1). The time lag introduced in Experiment 3 has been overcome by training the network for a concatenated audio–video input. The modified encoder is shown in Fig. 3. Modified decoder is just a mirror image of this new encoder. Together, they form the second type of modified autoencoder proposed in this paper. The first hidden layer implements fusion along with dimensional reduction, whereas the second hidden layer will focus on reducing this fused feature space into a latent space. The MLP was trained as usual. This case is similar to showing video with audio in sync. The participants in our example will be able to correlate better between the two modalities. The results of all four experiments have been shown in the next section.

4 Results

4.1 Experiment 1—Using Audio-Only Features as Inputs

A gradual decrease in reconstruction error was observed in this experiment. It supports the idea of greedy layer-wise pre-training for deeper layers. It is also observed that for the same architecture used the reconstruction error for arousal dimension is much lower than the valence dimensions. This is further justified by observing the mean square error during the prediction in the regression phase. Figure 4 shows a comparison between the arousal and valence dimensions for regression phase (validation set during training). Clearly, as stated in the baseline paper [20], the audio is more suitable to detect arousal values when compared to valence values.

Table 1 shows a comparison of valence and arousal values for audio-only input. The training was terminated once the mean square error stopped decreasing in steps of 0.002 for more than 5 iterations.

4.2 Experiment 2—Using Video-Only Features as Input

The custom-built layer was able to produce a better reconstruction than the usual dense layer used in the audio-only input. As mentioned in [20] while capturing the video, the face was not detected for some time frames. It was tested using only dense layer instead of custom-built layer, but the reconstruction error was larger than that observed for this case. The reason is—autoencoders had difficulties in reproducing zero as output for a particular dimension if the input for the same dimension was zero. This was contributing to increase the reconstruction error in case when the dense

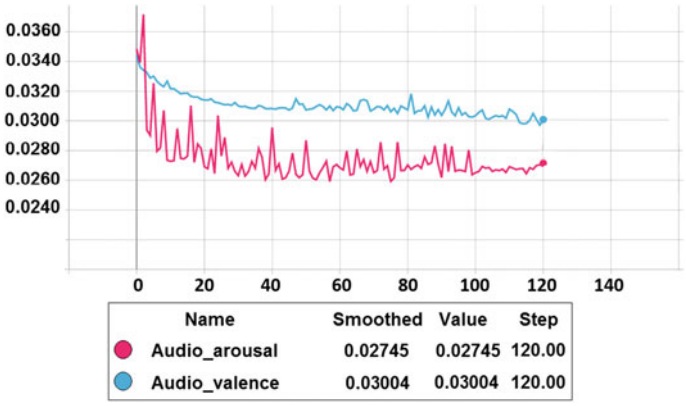


Fig. 4 Mean square error (validation) versus iterations for autoencoder with audio-only input

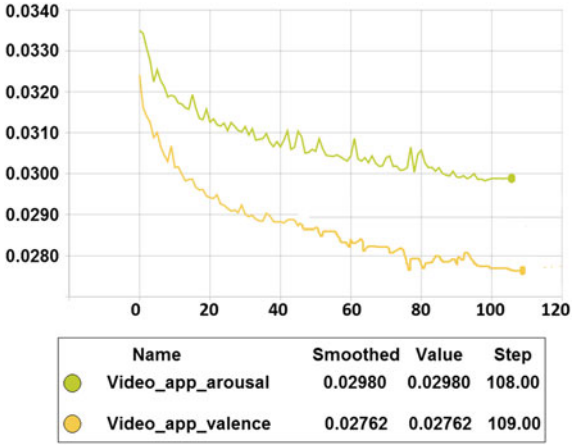


Fig. 5 Mean square error (validation set) versus iterations for autoencoder with video-only input

layers were used. This problem was solved using the custom-built layer explained in Sects. 2.1 and 2.2. Similarly, for video—valence dimensions—a lower reconstruction error and lower mean square error than audio were observed. Figure 5 shows the graphs for the same.

4.3 Experiment 3—Using Sequential Audio and Video Input

This experiment uses both the modalities but in a sequential manner. First, the video modality is presented as the input for training the network, and then the audio modality is presented to the same network. As explained earlier while capturing the video for some frames, complete face was not detected. So, the features for these frames in the dataset were nullified or were brought down to zero. The emotional dimensions were not being identified in this case, but with audio inputs coming after the video in sequence the network was able to identify emotional dimension for those frames as well. This compensates for the missing modality but with lag in time. The actual loss includes readings from the time frames when one of the modalities was erroneous. Figure 6 shows the mean square error for this setup. Table 1 shows a comparison of arousal and valence values for sequential audio–video input.

4.4 Experiment 4—Using Concatenated Audio and Video Input

This was the final experiment with concatenated audio and video modalities. Fig. 7 shows the mean square error versus iterations for arousal and valence values in this

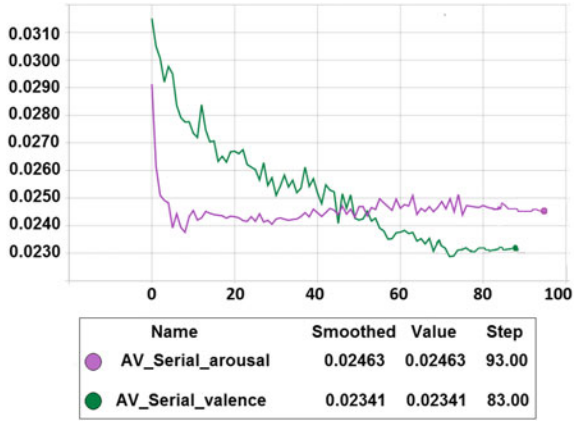


Fig. 6 Mean square error (validation set) versus iterations for audio–video sequential input—arousal and valence

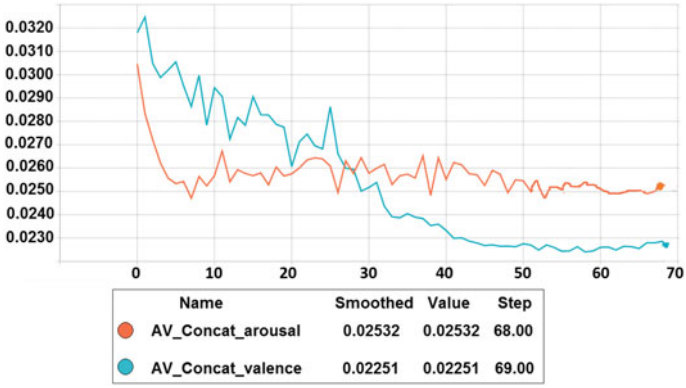


Fig. 7 Mean square error (validation) versus iterations for concatenated audio–video input—arousal and valence

Table 1 Comparison—mean square error for arousal and valence values in different experiments

RECOLA dataset	(a) Audio only		(b) Video only		(c) Audio–video sequential		(d) Audio–video concatenated	
	Arousal	Valence	Arousal	Valence	Arousal	Valence	Arousal	Valence
Train	0.0210	0.0240	0.0217	0.0235	0.0162	0.0143	0.0189	0.01419
Validation	0.0274	0.0300	0.0298	0.0276	0.0246	0.0234	0.0253	0.02251
Test	0.0281	0.0318	0.0319	0.0296	0.0258	0.0235	0.0255	0.02288

case. The network has access to both modalities at the same time and can find correlations and hence predict emotional dimensions with a better accuracy as compared with other experiments. Table 1 shows the comparison in all the four experiments and their results.

Table 2 Comparison—root-mean-square test set error for arousal and valence values on different approaches using various feature sets available in RECOLA dataset. The asterisk denotes baseline results computed by organizers of AV + EC 2015

Technique used	Feature sets used from RECOLA dataset	Arousal	Valence
Support vector regression and neural networks [18]	Audio, video (landmark, LGBP-TOP), ECG, EDA	0.164	0.113
Random forest [21]	Audio, video (HOG, LGBP-TOP, and PHOG-TOP), ECG, EDA	0.187	0.139
Fixed mapping [21]	Audio, video (HOG, LGBP-TOP, and PHOG-TOP), ECG, EDA	0.159	0.138
Gradient boosting [21]	Audio, video (HOG, LGBP-TOP, and PHOG-TOP), ECG, EDA	0.175	0.124
LSTM + RNN [12]	Audio, video(landmark, geometric LGBP-TOP), face-CNN, ECG, EDA	0.137	0.103
Our approach	**Audio, video (LGBP-TOP features only)**	**0.155**	**0.142**

4.5 Comparison of Results

The RECOLA dataset has the following modalities available: one feature set for audio modality, three feature sets for video modality (LGBP-TOP/appearance, landmark, and geometric features), an ECG and an EDA feature set for physiological modality. In this paper, two of the above-mentioned modalities—audio and video (with LGBP-TOP feature set only)—have been used. Table 2 shows a comparison with the other multimodal approaches and architectures on the same dataset. The details of feature sets used and the approach are mentioned in Table 2. In [12], the authors use long short-term memory—recurrent neural networks-based architecture to predict emotional dimensions. They use audio feature set, all the three video feature sets (LGBP-TOP, landmark, geometric) along with a convolutional neural network (CNN)-based appearance features (face-CNN feature set). Further, the authors also use ECG and EDA feature sets to obtain the results shown in Table 2. The organizers of Audio-Visual Emotion Recognition Challenge 2015 have released their baseline results on RECOLA dataset in [18]. Several robust ensemble methods have also been used for emotion recognition. In [21], the authors propose ensemble methods like random forest, gradient boosting for robust regression models. Their results using audio, video (HOG, LGBP-TOP, and PHOG-TOP), and physiological feature sets have been shown in Table 2. Modified SAEs used in this paper are computationally less expensive than CNN- and RNN-based techniques and achieve comparable performance with only two such features sets (audio and video using LGBP-TOP). The proposed architecture is simple to implement, outperforms the ensemble-based methods for arousal dimension, and achieves comparable results in valence dimensions.

5 Discussion and Conclusion

Autoencoders, commonly used to efficiently reduce dimensions of the input vector, can also be effectively used to fuse information from multiple modalities. Modified autoencoders are used to map the input dimensions of various modalities to a common latent space, which represents the information in a fused form. The results in this paper illustrate that the fused form of information identifies emotional dimensions in a more efficient way than single modality. If one of the modalities is corrupted, the other modality still compensates for the loss of information due to the corrupted one. The autoencoders when compared to the other architectures give a cost-effective way to fuse information. The architecture turns out to be simple to implement and is efficient in terms of computations and memory consumption when compared to the other approaches using convolutional neural networks and recurrent neural networks. The achieved performance is better or at par with other techniques (after using only two modalities). This approach may find its use in smart medical devices which can be used in future for medical diagnosis.

References

1. Fujisaki, W., Goda, N., Motoyoshi, I., Nishida, S., Komatsu, H.: Audiovisual integration in the human perception of materials. J. Vis. **14**, 1–20 (2014)
2. Khorrami, P., Paine, T.L., Brady, K., Dagli, C., Huang, T.S.: How deep neural networks can improve emotion recognition on video data (2016). arXiv:1602.07377
3. Zhang, S., Zhang, S., Huang, T., Gao, W.: Multimodal deep convolutional neural network for audio-visual emotion recognition. In: Proceedings of the 2016 ACM International Conferences on Multimedia Retrieval, ICMR'16, pp. 281–284 (2016)
4. Wu, Z., Sivadas, S., Tan, Y.K., Bin, M., Goh, R.S.M.: Multi-modal hybrid deep neural network for speech enhancement (2016). arXiv:1606.04750
5. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the 28th International Conferences on Machine Learning, pp. 689–696 (2011)
6. Uzan, L., Wolf, L.: I know that voice: Identifying the voice actor behind the voice. In: Proceedings of the 2015 International Conferences on Biometrics, ICB'15, pp. 46–51 (2015)
7. Salah, A.: Perceptual information fusion in humans and machines. Cogn. Neurosci. Forum. (2007)
8. Yang, L., Jiang, D., He, L., Pei, E., Oveneke, M.C., Sahli, H.: Decision tree based depression classification from audio video and language information. In: Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge, AVEC'16, pp. 89–96 (2016)
9. He, L., Jiang, D., Yang, L., Pei, E., Wu, P., Sahli, H.: Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks. In: Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge, AVEC'15, pp. 73–80 (2015)
10. Fontaine, J.R.J., Scherer, K.R., Roesch, E.B., Ellsworth, P.C.: The world of emotions is not two-dimensional. Psychol. Sci. **18**, 1050–1057 (2015)
11. Breazeal, C.: Emotion and sociable humanoid robots. Int. J. Hum Comput Stud. **59**, 119–155 (2003)
12. Chao, L., Tao, J., Yang, M., Li, Y., Wen, Z.: Long short term memory recurrent neural network based multimodal dimensional emotion recognition, pp. 65–72 (2012). arXiv:1212.5701

13. Ringeval, F., Sonderegger, A., Sauer, J., Lalanne, D.: Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. In: 2013 10th IEEE International Conferences on Workshops on Automatic Face and Gesture Recognition (2013)
14. Keras Documentation. https://keras.io/
15. TensorFlow. https://www.tensorflow.org/
16. Wang, W., Ooi, C., Yang, X., Zhang, D., Zhuang, Y.: Effective multi-modal retrieval based on stacked auto-encoding. Proc. VLDB Endow. **7**, 649–660 (2014)
17. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
18. Ringeval, F., Schuller, B., Valstar, M., Jaiswal, S., Marchi, E., Lalanne, D., Cowie, R., Pantic, M.: Av+Ec 2015—The first affect recognition challenge bridging across audio, video, and physiological data. In: Proceedings of the 5th International Workshops Audio/Visual Emotion Challenge, AVEC'15, pp. 3–8 (2015)
19. Zeiler, M.D.: ADADELTA: an adaptive learning rate method (2012). arXiv:1212.5701
20. Valstar, M., Pantic, M., Gratch, J., Schuller, B., Ringeval, F., Lalanne, D., Torres Torres, M., Scherer, S., Stratou, G., Cowie, R.: AVEC 2016—depression, mood, and emotion recognition workshop and challenge. In: Proceedings of the 6th International Workshops Audio/Visual Emotion Challenge, AVEC'16, pp. 3–10 (2016)
21. Kächele, M., Thiam, P., Palm, G., Schwenker, F., Schels, M.: Ensemble methods for continuous affect recognition, pp. 9–16 (2012). arXiv:1212.5701

Autoencoding Convolutional Representations for Real-Time Eye-Gaze Detection

Tharun Kumar Reddy, Vinay Gupta and Laxmidhar Behera

Abstract Several real-life applications, like driver drowsiness detection systems, lie detector systems (eye accessing cues), are based on gaze direction of human eyes. Estimated gaze direction is an indication of user's region of focus in space. In this work, we present a real-time eye-gaze detection pipeline on Eye Chimera database. Facial landmarks are used to detect face regions accurately, and we do provide a custom modification of Viola-Jones algorithm for eye region localization. The obtained eye regions dataset is fed to CNN models for training. Further, the obtained CNN features from the trained models have been fused using an autoencoder. This idea is not much explored in the literature which was much recommended in the AlexNet paper by Krizhevsky et al. 2012. Results demonstrate statistically significant ($p < 0.01$) improved classification performance than the recently proposed methods on this database.

Keywords Drowsiness detection · Autoencoder · Gaze direction

1 Introduction

Over the years, eye-gaze tracking spanned a wide number of applications, like driver drowsiness detection, human–computer interaction, medical diagnoses [1], gaming, virtual reality, Web advertisements, and various psychological studies [2]. Develop-

T. K. Reddy (✉) · L. Behera
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, Uttar Pradesh, India
e-mail: tharun@iitk.ac.in

L. Behera
e-mail: lbehera@iitk.ac.in

V. Gupta
Department of Mathematics and Statistics, Indian Institute of Technology Kanpur, Kanpur 208016, Uttar Pradesh, India
e-mail: vinayg@iitk.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_18

ments in computing speed, digital video processing and availability of cheap hardware are the chief bottlenecks which are being surpassed in post-2000 era which are making gaze tracking user friendly. Online eye-gaze and eye-state tracking in automotive industry are used in driver assistance systems to evaluate driver vigilance and sleepiness levels. For instance, a computer vision gaze-based tracking system in a car uses a gaze tracker device or a camera mounted on dashboard with the computing hardware processing the machine learning pipeline with processed output cues given to awaken the driver. Research also provides access to interactive gaze-based displays, eye-gaze-assisted functionalities in smartphones [3, 4]. Within each of the user applications, there exists a large variation in system configurations, operating environment, and imaging equipment. The challenges of reproducibility in the adaptation of gaze tracking performance for new subjects are still a challenging issue. Thus, after several decades of research in eye-gaze tracking, the problem of performance comparison among such system variations is still an issue [5].

In the literature, gaze tracking methods are grouped into five major classes: polynomial regression-based [6–10], 3D model-based [11–13], cross-ratio-based [14, 15], appearance [16, 17], and shape-based methods [18, 19]. The first three methods constitute corneal reflection-based ones which throw near-infrared light to estimate gaze direction using polynomial regression or geometrical model of the eye.

Polynomial regression strategies for gaze detection employ various characteristics of the eye, like geometry, eye pupil silhouettes, corneal reflections, and are grossly deployable with the help of a camera and some near-infrared LEDs. Gaze detection accuracy of these techniques is limited by head movements. Head movements rapidly decrease the accuracy of the polynomial regression algorithms; therefore, it requires enforcing user inconvenience by holding the head, chin, or mouth in metallic bars [6, 8]. 3D model-based techniques allow for user head motion, but the additional bottleneck exists in the form of costly hardware required, for instance: more than one camera, multiple luminance emitters [13], and a proper calibration. In contrary, the cross-ratio-based methods do not need a 3D model, no user inconvenience by restricting head motions, and no device calibration. Appearance model-based methods do not use external light shining sources. In general, they utilize eye contour's shape, surface texture, relative shift of pupils in comparison with contour corners in order to extract gaze direction. The devices used in such methods are also low-cost, low-resolution cameras without any supplementing light source. The performance of these models could be improved by collecting large training data encompassing various environments. Shape-based techniques fit various deformable shape patterns for the eye contours and iris and ultimately unify them for fitting to an aggregate eye appearance. These models produce good accuracies even with low-resolution camera images. But the drawbacks constitute complexities in fixing templates for varying head poses, eyebrow occluding the iris, model adaptability to various eye shapes in subjects, computational complexity, appropriate starting model. In this work, we primarily contribute to the appearance model-based methods, in the context of utilizing low-cost webcams without many light sources. Within the scope of these methods, in past few years, convolutional neural network-based methods are proposed for gaze estimation. In a recent paper [20], authors classify the eye-gaze

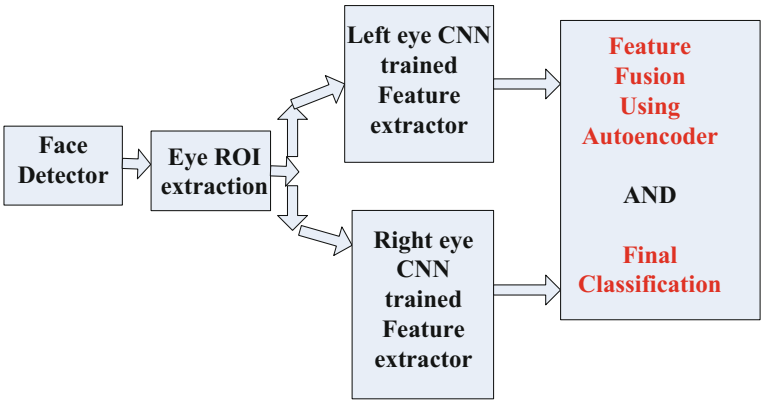


Fig. 1 Block diagram of the procedural pipeline

directions using a convnet with three convolutional layers based on data collected from webcams with no calibration. We re-implement this algorithm as one of the baseline methods. Several other papers in these methods include [21], where authors describe a near-eye display control using gaze tracking which is robust to illumination environment, skin, pupil, skin color built using CNN's. In [23], authors build a smartphone application which is used to collect images from 1450 people under varying illuminations, background, head pose, and a CNN eye-gaze tracking model is trained on the cropped faces, eyes from these people.

The paper proposes a real-time pipeline for eye-gaze direction prediction, tested under free natural light. Modified Viola-Jones method for eye detection and autoencoder-based CNN feature fusion are the two major contributions in this work. The algorithm pipeline thus obtained is used for the first time, and it enhanced the state of the art on the eye-gaze detection problem. Viola-Jones method for eye detection is custom modified to enhance the eye detection performance. Convolutional neural network models are trained from scratch on the given dataset and further used for extracting feature representations. In addition, autoencoder-based fusion of convolutional feature representations from both left and right eyes is used before the classification process. Average processing time for the whole pipeline is 30 ms per frame on an i7 Intel, 8 GB RAM desktop system.

2 Proposed Method

The overall structure for the method is depicted in Fig. 1. Various individual blocks of the procedure are indicated below.

2.1 Face and Eye Detection

The first block in the gaze direction method comprises the face, eye ROI detection. Two methods are used. First one is based on facial landmarks-based detection, and second one is a custom modified version of Viola-Jones algorithm.

2.1.1 Facial Landmark Detection (ERT)

Given an input image, facial landmark detection tries to localize key points among the facial structures on face. The method implemented in the dlib library is based on the paper [22]. Although there are a variety of facial landmark detectors, essentially focused on localizing and labeling the facial regions, we use only the obtained landmark coordinates and extract eye ROIs using linear transformation over the coordinates at the eye corners.

2.1.2 Custom Viola-Jones Eye Detector (CVJED)

Due permission is obtained for the usage of eye-chimera database [24] from the owners via a Data Release Agreement. The detection module described in Fig. 2 consists of a face detector which is based on Viola-Jones method [28], and the obtained face ROI ($height, h_{face}$) is subjected to further processing. The top portion ($\frac{h_{face}}{5}$) of the obtained face is removed from the ROI, and then the remaining ROI is divided into three equal parts. The upper one-third portion is fed as input to the Haar Cascade for eye region detection. The accuracy of such a system is obtained to be close to 87%. Here, a correctly classified eye ROI contains both IRIS, eyebrows excluding portions of eye corners. The obtained final eye ROIs are re-sized to 250 × 250 using image super-resolution CNNs without loss of much resolution.

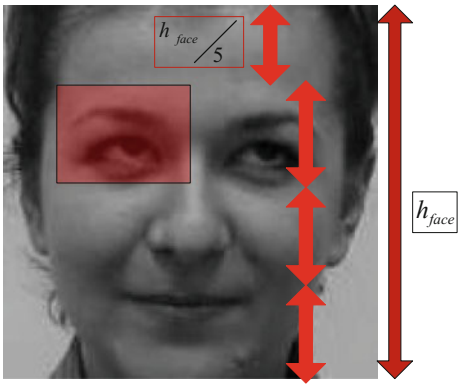


Fig. 2 CVJED algorithm on an image from Eye Chimera database [24]

The second block consists of a gaze direction classifier from the detected eye regions. CNN models are trained individually on both the eyes. The contribution of the proposed method lies in fusing the CNN features using an autoencoder.

2.2 Training of CNN Models and Feature Fusion, Classification

The eye ROIs obtained from left and right eyes are passed to individual convolutional neural networks for training. The key motivations for using convolutional neural networks are: local weight connections, shared weights, pooling, and multiple layers of training. Hence, they are the state of the models for processing data in the form of numerous arrays like images. Krizhevsky et al. [29] trained CNN on GPUs for classification on a large dataset like ImageNet. The accuracies obtained by CNNs are far superior to any conventional feed-forward neural network. Input consists of images of size 250×250. The CNN model used for training is shown in Fig. 3. It consists of first convolutional layer with 32 filters of size (3×3), stride 1, padding 1. It is followed by a rectified linear activation and a max-pooling operation. Max-pooling consists of spatial downsampling by a factor of 2 along both x and y directions. Two more layers follow the same pattern of parameters as above. Further, the network consists of 64 fully connected units in the pen-ultimate layer and a final layer of 7 units for classification. The network minimizes multiclass cross-entropy cost function. The network is trained using Adagrad optimizer for 11000 iterations. It took around 5 days on a single NVIDIA Titan X GPU. The CNN models training on both right and left eye regions are further used as feature extractors, and the features in the fully connected layer are extracted from both the CNN models (left, right). The

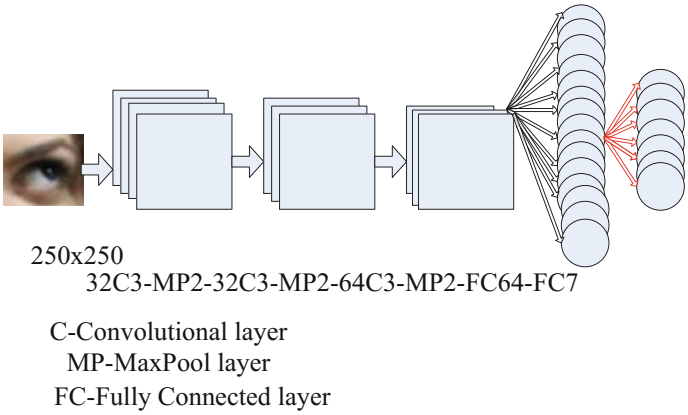


Fig. 3 Convolutional neural network model used

obtained features are fused using a standard autoencoder of size 96 × 64. The fused output features of size 64 × 1 are used for subsequent classification.

3 Experiments

All the results presented in the manuscript are conducted on the Eye Chimera database [24, 25] which contains webcam images of subjects with seven classes of gaze directions. The images are recorded using Canon 600D (640 × 480) and Panasonic HDC-TM 60 (1920 × 1080) cameras from 40 subjects in the 20–30 age range. The cameras were mounted on a stand. The database consists of 1170 frontal face images, arranged within seven classes. The database has been partitioned as per twofold cross-validation. Half of the randomly chosen samples were used for eye region localization which are further used for the training of CNN models. Other half was used for testing.

3.1 Procedure

Convolutional neural networks in general are data hungry models and require a huge number of examples to train and also generalize well on the test dataset. In comparison with the datasets like ImageNet, this is a considerably small one for training CNN models. Hence, the examples from the database are augmented with various images transformed by translation, rotation, blurring, noise addition, scaling. This increased the size of the dataset. CNN models were trained on the left and right eye regions on a NVIDIA deep learning dev-box 32 GB RAM, Intel 6 Core I75930K 3.5 GHZ 15M processor, with four NVIDIA GeForce Titan X GPUs. Further, the features in the fully connected layers of the trained CNN models are fused through an autoencoder and further trained supervisedly with class labels. Both the problems of three class and seven class gaze detection are considered in this work.

3.2 Results

Table 1 contains a performance comparison of the proposed method with state-of-the-art methods Anjith (ROI+CNN) [20], Anjith (ERT+CNN) [20]. Anjith (ROI+CNN) used a geometric calculations-based eye ROI detection algorithm, while Anjith (ERT+CNN) [20] used a facial landmark-based detection method for eye coordinates. In both of the above methods, the author used averaging at left and right eye CNN outputs for training and classification. Trained CNN models are under-performing as the author does not use fusion at feature level but rather uses a decision-level fusion (score averaging) from both the eyes for gaze direction classi-

Table 1 Per class classification accuracy performance comparison

Method	VD	VR	VC	AR	AC	ID	K
Anjith (ROI+CNN) [20]	96	79	87	77	79	75	93
Anjith (ERT+CNN) [20]	97	93	94	84	87	71	91
Proposed (CVJED+CNN+autoencoder)	97	83	87	84	80	77	90
Proposed (ERT+CNN+autoencoder)	98	95	94	88	89	89	93

Table 2 Performance comparison using one of the eyes

Eye ROI localization method	Gaze detection method	Accuracy (seven classes)	Accuracy (three classes)
BoRMaN point detector [26]	Isophote curvature [27]	32.00	33.12
Zhu [9]	Zhu [9]	39.21	45.57
Laura [24]	Laura [24]	48.64	78.57
ERT [20]	CNN	86.81	96.98
Proposed (CVJED)	CNN	90.10	95.12
Proposed (ERT)	CNN	92.35	97.20

fication. In the proposed methods, proposed (CVJED+CNN+autoencoder) and proposed (ERT+CNN+autoencoder) data from both the eyes are used for training CNN models and online feature fusion is accomplished rather than a decision-level fusion; this accounts for better classification accuracies in each of the columns. The results for all the classes using the proposed methods are found to be statistically significant in comparison with Anjith (ROI+CNN) [20] and Anjith (ERT+CNN) [20] using t test ($p < 0.01$). Table 2 contains a performance analysis of the algorithms using data from a single eye. The CNN-based algorithms ERT [20], proposed (CVJED), proposed (ERT) demonstrate superior performance for both the seven classes and three classes scenarios. The trend of results also shows that the performance of CNN models is good on three classes in comparison with seven classes. In this paper, experiments have been conducted using two methods CVJED and ERT for extracting eye region of interest. Data has been partitioned into two sets, 50% each for training and testing. The individual CNNs are trained with Keras data generator with augmentation. The trained CNNs act as feature extractors, and instead of ensembling the CNNs at the output layer, we fuse the CNN features from each model using an autoencoder 96×64. The motivation for this comes from the suggestion offered in the seminal paper [29]. The subsequent improvement in the performance is seen clearly in both Tables 3 and 4 over the results in the previous works [20, 24].

Table 3 Confusion matrix for three classes (chimera dataset)

		Predicted label		
		Centre	Right	Left
True label	Centre	**1.00**		
	Right	0.04	**0.96**	
	Left			**1.00**

Table 4 Confusion matrix for seven classes

		Predicted label						
		Centre	UpRight	UpLeft	Right	Left	DownRight	DownLeft
True label	Centre	**0.97**	0.01	0.01	0.01			
	UpRight	0.01	**0.95**		0.03	0.01		
	UpLeft			**0.94**		0.06		
	Right	0.05			**0.88**		0.07	
	Left			0.04		**0.89**		
	DownRight				0.21		**0.89**	
	DownLeft					0.05	0.02	**0.93**

4 Discussion

This paper proposes a real-time eye-gaze recognition pipeline which incorporates a modified Viola-Jones eye detector and a CNN feature-fused autoencoder. The major contribution of the paper lies in developing a modified Viola-Jones detector and an autoencoder-based CNN feature fusion architecture in Keras deep learning library. The performance of the proposed pipeline is established in this paper as outperforming the state-of-the-art results which uses CNNs with decision-level fusion. This is visible from Tables 2 and 3. The accuracy of classification in different gaze directions is improved with the proposed model. Specifically, the classification errors noticed in the upright eye gaze have been reduced. Significant occlusion of the iris by the eyelid is still a pertinent problem in vertical gaze direction. Future research should use RGB-D images to detect occlusion edges and do better feature processing and hence improve further the classification performance and decrease the processing time.

5 Conclusion

In this work, we present a real-time eye-gaze system. The obtained eye gaze is useful to extract cognitive states from NLP literature and also to give information about the state of the driver driving a car. The algorithm computations are barely minimum

once the pre-trained CNN models are used and could be deployed on a smartphone for better interactive applications and also to increase mobility for motor-disabled subjects.

References

1. Rayner, K.: Eye movements in reading and information processing: 20 years of research. Psychol. Bull. **124**(3), 372 (1998)
2. Holzman, P.S., Proctor, L.R., Levy, D.L., Yasillo, N.J., Meltzer, H.Y., Hurt, S.W.: Eye-tracking dysfunctions in schizophrenic patients and their relatives. Arch. Gener. Psychiatry **31**(2), 143–151 (1974)
3. Bazrafkan, S., Kar, A., Costache, C.: Eye gaze for consumer electronics: controlling and commanding intelligent systems. IEEE Consum. Electron. Mag. **4**(4), 65–71 (2015)
4. Wood, E., Bulling, A.: Eyetab: model-based gaze estimation on unmodified tablet computers. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 207–210. ACM (2014)
5. Kar, A., Corcoran, P.: Towards the development of a standardized performance evaluation framework for eye gaze estimation systems in consumer platforms. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE (2016)
6. Blignaut, P.: Mapping the pupil-glint vector to gaze coordinates in a simple video-based eye tracker. J. Eye Mov. Res. **7**(1) (2013)
7. Zhu, Z., Qiang J., Kristin, P.: Nonlinear eye gaze mapping function estimation via support vector regression. In: 2006 18th International Conference on Pattern Recognition, ICPR'06, vol. 1. IEEE (2006)
8. Brolly, X.L.C., Jeffrey, B.M.: Implicit calibration of a remote gaze tracker. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop, CVPRW'04. IEEE (2004)
9. Zhu, Z., Ji, Q.: Eye and gaze tracking for interactive graphic display. Mach. Vis. Appl. **15**(3), 139–148 (2004)
10. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2879–2886. IEEE (2012)
11. Wang, K., Ji, Q.: Real time eye gaze tracking with Kinect. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 2752–2757. IEEE (2016)
12. Zhou, X., Cai, H., Shao, Z., Yu, H., Liu, H.: 3D eye model-based gaze estimation from a depth sensor. In: 2016 IEEE International Conferences on Robotics and Biomimetics, Qingdao, pp. 369–374 (2016)
13. Jianfeng, L., Shigang, L.: Eye-model-based gaze estimation by RGB-D camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 592–596 (2014)
14. Coutinho, F.L., Morimoto, C.H.: Augmenting the robustness of cross-ratio gaze tracking methods to head movement. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 59–66. ACM (2012)
15. Hansen, D.W., Agustin, J.S., Villanueva, A.: Homography normalization for robust gaze estimation in uncalibrated setups. In: Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications, pp. 13–20. ACM (2010)
16. Wu, Y.L., Yeh, C.T., Hung, W.C., Tang, C.Y.: Gaze direction estimation using support vector machine with active appearance model. Multimed. Tools Appl. **70**(3), 2037–2062 (2014)
17. Zhang, X., Sugano, Y., Fritz, M., Bulling, A.: Appearance-based gaze estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4511–4520 (2015)

18. Wang, W., Huang, Y., Zhang, R.: Driver gaze tracker using deformable template matching. Proceedings of the IEEE International Conferences on Vehicular Electronics and Safety, ICVES'11, pp. 244–247 (2011)
19. Ince, I.F., Kim, J.W.: A 2D eye gaze estimation system with low-resolution webcam images. EURASIP J. Adv. Signal Process. **2011**(1), 40 (2011)
20. George, A., Routray, A.: Real-time eye gaze direction classification using convolutional neural network. In: 2016 International Conference on Signal Processing and Communications (SPCOM). IEEE (2016)
21. Konrad, R., Shikhar, S., Varma, P.: Near-Eye Display Gaze Tracking via Convolutional Neural Networks
22. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874 (2014)
23. Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., Torralba, A.: Eye tracking for everyone. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2176–2184 (2016)
24. Florea, L., Florea, C., Vrnceanu, R., Vertan, C.: Can your eyes tell me how you think? A gaze directed estimation of the mental activity. In: BMVC, January 2013
25. Vrnceanu, R., Florea, C., Florea, L., Vertan, C.: NLP EAC recognition by component separation in the eye region. In: International Conference on Computer Analysis of Images and Patterns, pp. 225–232. Springer, Berlin, Heidelberg (2013)
26. Valstar, M., Martinez, B., Binefa, X., Pantic, M.: Facial point detection using boosted regression and graph models. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2729–2736. IEEE (2010)
27. Valenti, R., Gevers, T.: Accurate eye center location and tracking using isophote curvature. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR'08, pp. 1–8. IEEE (2008)
28. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'01, vol. 1. IEEE (2001)
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)

Hausdorff Distance-Based Binary Search Tree Multiclass Decomposition Algorithm

Rahul Kumar Sevakula and Nishchal K. Verma

Abstract Multiclass decomposition algorithms are the means by which binary classification algorithms, like support vector machine, are used for multiclass classification problems. The popular multiclass decomposition algorithms, like one against one (OAO), one against all (OAA), perform the decomposition in a naive manner. This paper presents a novel heuristic-based decomposition algorithm that takes the Hausdorff distance between two classes to decide the decomposition. The presented algorithm has been evaluated and compared against OAO and OAA methods across nine datasets. The comparison shows that presented method not only provides comparable performance, but also in most cases can classify the test samples with fewer average number of support vectors, thus leading to faster test performance.

Keywords Multiclass classification · Decomposition algorithms · Support vector machine · Hausdorff distance

1 Introduction

Binary classifier has a very special place in the history of classification problems. The first classifiers were prominently binary classifiers, e.g., linear discriminant analysis. The classifiers having maximum ability to achieve low generalization error are also binary classifiers, e.g., SVM and MVP classifiers [1, 2]. When we look back in history, many of the classifiers were first designed for binary classification and later extended to multiclass classification [3]. Primarily, there have been two ways by which binary classifiers are made capable of solving multiclass classification problems: (1) extend the learning algorithm to a multiclass version and (2) decompose the

R. K. Sevakula (✉)
IBM India, Bengaluru 560071, India
e-mail: rahul.sevakula@gmail.com

N. K. Verma
Indian Institute of Technology, Kanpur, Kanpur 208016, India
e-mail: nishchal@iitk.ac.in; nishchal.iitk@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_19

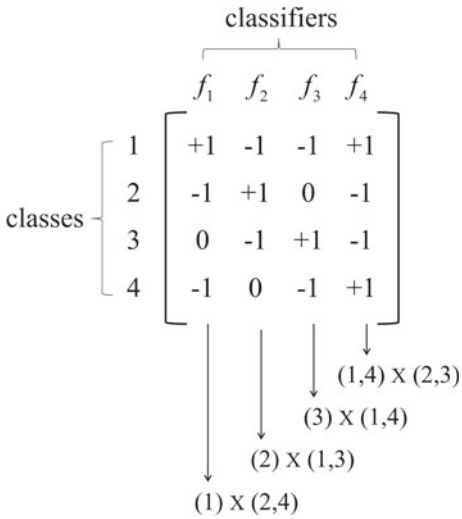
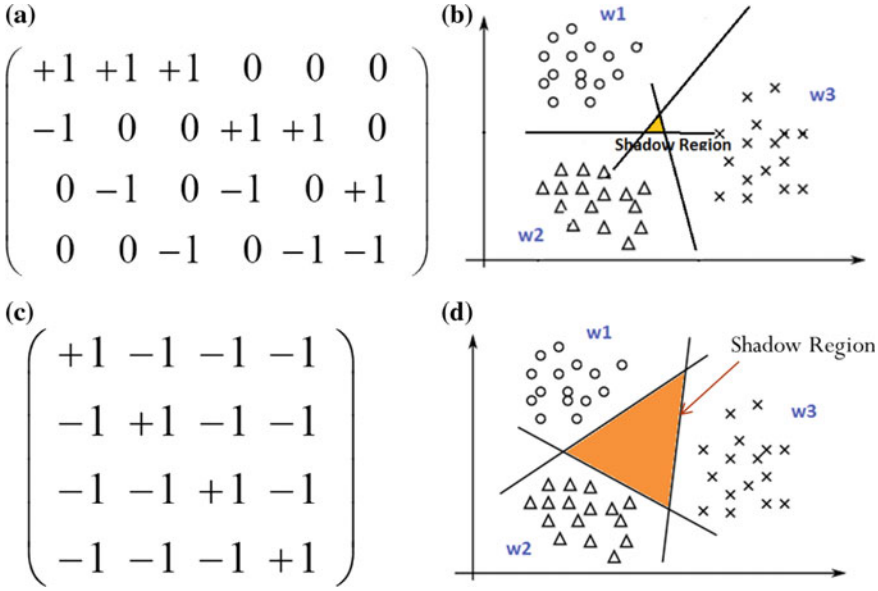


Fig. 1 Code-matrix framework to describe multiclass decomposition algorithms

multiclass problem into binary subproblems. The first method can lead to computationally costly algorithms [3] or sometimes may also be impractical to formulate. The second method on the other hand is easy to implement and provides the facility of parallel processing by independently solving the binary subproblems using different processors. This paper shall therefore focus on the second class of methods.

Lorena et al. [4] present an excellent literature survey on decomposition algorithm in this context. Decomposition algorithms involve two steps: (1) dividing the multiclass problem into binary subproblems and (2) combining the results of the binary classifiers to assign a class to the sample. Allwein et al. [5] proposed a code-matrix framework, represented by code matrix **M** for getting more insight into the decomposition algorithms. Consider a multiclass classification problem having k classes which is decomposed to l binary subproblems. The columns define the labeling that the k classes assume while training individual binary classifiers. Thus, **M** has the dimensions of $k \times l$. Element $m_{i,j}$ gives the expected value of jth classifier w.r.t. ith class and may assume the values of $-1, 0, +1$. '+1' indicates positive label, '−1' indicates negative label, and '0' indicates non-participation from class i. Figure 1 shows a sample code matrix. One can observe that given a value of k, there is no restriction on number of classifiers. This means the codes for individual classes can be either made with lot of redundancy or can be made very compact. The most compact decomposition possible for a multiclass classification problem of k classes would be $l = log_2(k)$.

The most popular of all decomposition methods is one against one (OAO) and one against all (OAA). Sometimes, these methods are also referred to as one versus one (OVO) and one versus all (OVA), respectively. OAO decomposes the multiclass classification problem to a situation where a binary classifier is trained to classify each possible pair of individual classes. Therefore, the number of binary classifiers trained



$$\begin{pmatrix} +1 & +1 & +1 & 0 & 0 & 0 \\ -1 & 0 & 0 & +1 & +1 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$$

Fig. 2 Illustration: **a** code matrix for OAO, **b** illustration for OAO, **c** code matrix for OAA, and **d** illustration for OAA

would equal the number of possible combinations that two classes could be selected from k classes, which is equal to $\frac{k(k-1)}{2}$ binary subproblems. Final assignment of class is typically done by taking a majority vote across decisions taken by the individual classifiers. A major disadvantage faced by OAO is the presence of shadow region, which is a region where final class assignment may turn out to be a tie. OAA on the other hand solves the problem by training a binary classifier for each class versus all remaining classes together, leading to a total of k classifiers. The individual classifiers classify the test samples with a probability describing the belongingness of the sample to the individual classifier. Final assignment of class is typically done by finding the class that was assigned maximum probability. A disadvantage of OAA approach is that typically each of the binary classifiers is to be trained on class-imbalanced data (as classifier is learned between data of one class and the data of all other classes together). Accordingly, class imbalance methods may be required to be employed while training each of the classifiers. Figure 2a–d presents the code matrix and illustrations for OAO and OAA, respectively.

Dietterich and Bariki [6] proposed the use of error-correcting output codes (ECOC). The idea behind ECOC is to have more number of classifiers so that some redundancy is created [23, 26]. This redundancy provides the facility to correct errors. Allwein et al. [5] pointed out that although the codes generated by the ECOC decomposition have good error-correcting properties, many of the binary subproblems generated can be difficult to learn. For this reason, case studies showed that simpler decompositions, like OAA and OAO, often provide comparable and some-

times superior results as compared to ECOC. Further, ECOC is computationally not efficient as well [7]. Another popular set of strategies are hierarchical strategies [22, 27]. Some examples are Decision Directed Acyclic Graph (DDAG) [8] and Adaptive Directed Acyclic Graph [9]. A disadvantage of DDAG as pointed by Kijsirikul and Ussivakul [9] is that results are dependent on the sequence of binary classifiers and that there is possibility of shadow region being present. Another deficiency of DDAG is that the number of classifiers evaluated is dependent on the true class's position in the graph. For example, if correct class is already present in root node, it would still be evaluated with $k - 1$ binary classifiers before obtaining the output. This increases the probability of error as cumulative error keeps building up, which is critical for large databases. Cumulative error would be $1 - (1 - \alpha)^{k-1}$, when α is error rate per node. Directed binary trees [10–13] need only $k - 1$ classifiers. Further, in case of a balanced binary tree, only $log_2 k$ classifiers need to be used for classifying test samples; therefore, it has least amount of redundant in the code matrix. For the directed binary tree, partitioning of classes primarily involved the use of clustering methods [10–12]. Lorena and Carvalho [13] treated the decomposition as a graph and used minimum spanning tree for making the decomposition.

Some drawbacks faced in earlier methods for forming the directed binary tree (1) are that earlier methods usually end up giving us imbalanced binary trees; thus, the main advantage of fast test timing is not achievable; (2) most of the measures for calculating separation between classes fail in simple cases, e.g., sets made of concentric circles, sets having almost equal number of samples. This paper presents a novel decomposition algorithm which brings out a balanced binary tree decomposition of classes. For partitioning of the classes, Hausdorff distance (HD) is used to identify the classes which are most separable from each other. As these two most separable classes form two partitions, remaining classes are assigned to each partition based on their affinity to the partition w.r.t their Hausdorff distances. Special consideration ensures that a balanced binary tree-like structure is formed, which theoretically gives least redundant code matrix. Experimentation on nine datasets shows that proposed method achieves comparable classification performance with OAO and OAA. Further, the proposed method was shown to achieve the mentioned classification performance with least number of nonzero support vectors (SVs), which translates to faster test performance and lesser requirement of memory for storing the SVs. Further paper proceeds as follows: Sect. 2 explains Hausdorff distance, and Sect. 3 describes the proposed decomposition algorithm. Section 4 mentions the performed experimentation, results achieved, and conclusions derived from the same. Section 5 ends the paper with conclusions.

2 Hausdorff Distance

Hausdorff distance is a popular measure to find out how close two non-empty sets of a metric space are to each other [14–16]. Distance between two sets can be pictured as distance between a representative point from each set. It is important here that the

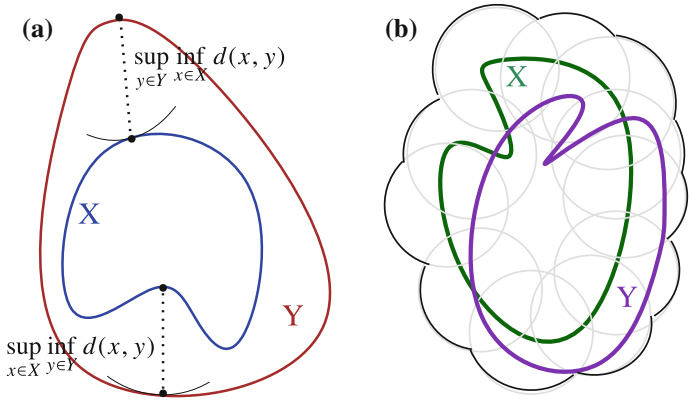


Fig. 3 Illustration to depict: **a** first definition of Hausdorff distance and **b** second definition of Hausdorff distance

two sets between whom Hausdorff distance is to be computed are non-empty sets. Closed and bounded properties of the space are very essential as it allows one to take limits. In the absence of such restrictions, it cannot be ensured that limit exists and if it exists, it will be finite. The restriction will also allow us to have $d(A, B) = 0$ when $A = B$, which is a necessary criterion for a metric space to be defined.

There exist two mathematical definitions for Hausdorff distance [14]. The first definition mentions that two sets are said to be close when every point of either set is close to some point in the other set. Given a compact metric space S, consider X to be space of non-empty closed subsets of S. Hausdorff metric is defined on pairs of elements in X whose expression is given in Eq. 1.

$$\begin{aligned} X &= \{A \subset S |\ A \text{ is non-empty, closed and bounded}\} \\ d_H(A, B) &= \max \left\{ \sup_{x \in A} \inf_{y \in B} d(x, y), \sup_{y \in B} \inf_{x \in A} d(x, y) \right\} \end{aligned} \tag{1}$$

The second definition which is graphically more appealing is defined as follows. Given $A \in X$, let its ϵ-expansion be defined as the union of all ϵ-open spheroids around points in A and denoted as $E_\epsilon(A)$. Hausdorff distance between the two sets is defined as the smallest ϵ that allows the expansion of one set to cover the other and vice versa [15]. The expression is given in Eq. 2, and its illustration is shown in Fig. 3.

$$\begin{aligned} E_\epsilon(A) &= \cup_{x \in A} B(x, \epsilon) \\ d_H(A, B) &= \inf \{\epsilon > 0 |\ E_\epsilon(A) \supset B \ and\ E_\epsilon(B) \supset A\} \end{aligned} \tag{2}$$

3 Proposed Method

The proposed method for decomposition is presented in Algorithm 1. The primary concept behind the method is to find the classes that are most distant from each other and then segregate other classes into two groups based on their proximity to each of these two classes. This process is recursively repeated to bring out a binary tree-like structure to divide the classes. Considering that it is very likely that each of the classes comprise of more than one training sample, metrics which can measure distance between two non-empty sets could be useful for our decomposition algorithm. Some popular options for this objective are: (1) Jaccard distance, (2) Earth mover's distance after estimating probability distributions for each of the two non-empty sets, and (3) the Hausdorff distance. Jaccard distance has a limitation that when the two sets are mutually exclusive, no matter how far or close the two sets be, the Jaccard distance would always be 1. This limits the ability to compare the distance between more than two mutually exclusive non-empty sets. The second option of finding Earth mover's distance after estimating probability distributions can a very computationally expensive procedure, as estimating probability distributions is not at all an easy task. On considering these options, the easy-to-compute Hausdorff distance appeared to be the most suitable metric here.

Algorithm 1: Proposed decomposition algorithm using Hausdorff Metric

---

Step 1: Calculate the Hausdorff distance of all classes from each other. Given k classes, in total we will have $\frac{k(k-1)}{2}$ pairs of classes for which Hausdorff distance is found.

Step 2: Make two empty sets of classes, namely $left_node$ and $right_node$ (the way we have in binary search trees). Find the pair of classes that are most distant from each other w.r.t. Hausdorff distance. From this pair, assign one class to $left_node$ and other to $right_node$.

Step 3: Now based on Hausdorff distances already found, we check the class, i.e., closest to either left node or right node and then assign it to the closer node.

Step 4: Step 3 is repeated until the left node or the right node have half of the total number of classes present in the previous node. In case, the previous node has an odd number of classes, half will be rounded off to the higher integer value. The remaining classes are automatically assigned to the node which has fewer number of classes.

Step 5: Once all the classes have been divided (equally) into left node and right node, we recursively perform the Steps 2–4, to further divide each of these nodes. This recursive operation continues until the leaf nodes have only one class in them.

Following the proposed method gives a balanced binary tree-like structure. To make a multiclass decision-making system, a binary classifier needs to be generated

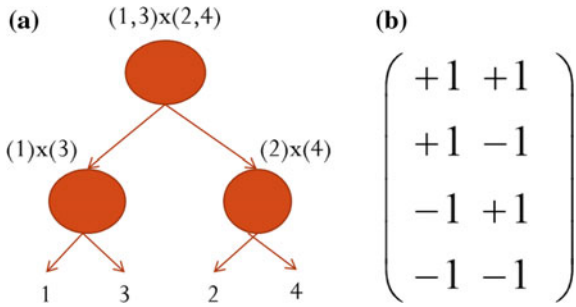


Fig. 4 **a** Partitioning into two groups of classes and **b** code matrix of proposed approach

for all the nodes except for leaf nodes, such that classes in left node and classes in right node are maximally separated. A pictorial presentation of how classes are divided into binary tree structure is shown in Fig. 4a, and the effective code matrix for number of classes, $k = 4$, is shown in Fig. 4b. During testing, starting from root node which has all classes, decision of binary classifier at each node will direct us to the next node. This operation continues until we reach a leaf node, which will give the determined/predicted class. An important point to be noted is that a case of shadow region, i.e., ambiguity during final class assignment does not exist with proposed approach.

4 Case Study and Results

To infer conclusions on the effectiveness of proposed method with respect to the traditional methods, nine datasets from UCI repository were tested upon. Details of the nine datasets are reported in Table 1. Using SVM as the binary classifier, the multiclass classification performance was compared across three decomposition methods: (1) one against one, (2) one against all, and (3) the proposed method. Further, each of the decomposition methods was evaluated upon the data in two cases/situations: (1) when data was normalized with zero-one normalization and (2) when data was not normalized. The case study presented in this paper closely follows the very famous paper by Hsu and Lin [3], which is also the most cited paper for comparing multiclass classification methods with SVM.

The first six datasets shown in Table 1 do not have a separate test set. Accordingly for these six datasets, classifier performance was evaluated based on tenfold cross-validation, and the best classification accuracy among the tenfold was reported. The remaining three datasets had a distinct training and test dataset; accordingly, classifiers were trained on the training data, and their performance was evaluated and reported on the test data.

All experiments were performed on MATLAB, and LibSVM [17] was used for training each of binary SVM classifiers. RBF kernel was used in all experiments. Following procedure was using for SVM parameter tuning in each of the experiments.

Table 1 Details of benchmark datasets

Datasets	Train samples	Test samples	Class	Features
Iris	150	0	3	4
Wine	178	0	3	13
Glass	214	0	6	13
Vowel	528	0	11	10
Vehicle	846	0	4	18
Segment	2310	0	7	19
Sat. image	4435	2000	6	36
Optical	3823	1797	10	64
Letter	16,000	4000	26	16

A total of 225 pairs of (C, γ) values were searched upon. The value for γ was logarithmically searched across powers of 2, from 2^{-10} to 2^4, and similarly C was searched across powers of 2, from 2^{-2} to 2^{12}. Fivefold cross-validation performance on the training data (considered at that stage) was found for each of the pairs, and the pair giving best performance was selected.

There are two measures across which performance of decomposition methods has been evaluated: (1) classification performance, (2) average number of nonzero support vectors required to pass through before determining the class. The decision function for SVM is given below where x refers to a test data sample, n refers to the number of training samples in training SVM, α^* refers to optimal value of SVs found after training, b^* refers to optimal value of b found with training, and $K()$ refers to the kernel function.

$$f(x, \alpha^*, b^*) = \sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b^* \tag{3}$$

The computational complexity in classifying test samples is directly and linearly proportional to the number of nonzero SVs [18, 19]; therefore, the second measure of average number of SVs indicates the speed at which classification of test samples would occur. The two measures as reported for each of the experiments are shown in Tables 2 and 3, respectively. In the tables, results with maximum classification accuracy have been reported in bold and results with lowest average SVs have also been reported in bold. From the results tables, it can be clearly ascertained that proposed approach brings about comparable performance with OAO and OAA approaches. Further, it is also observed that the average number of nonzero support vectors is always minimal in the case of the proposed approach. The likely reason for this observation is that a balanced binary tree structure has been ensured in our approach without compromising on observed classification performance. As expected, normalization has helped in having fewer number of nonzero SVs.

Table 2 Results—classification accuracies (in %)

Datasets	One against one		One against all		Proposed	
	After norm.	Without norm.	After norm.	Without norm.	After norm.	Without norm.
Iris	**99.33**	**99.33**	98.67	**99.33**	**99.33**	**99.33**
Wine	99.44	99.44	**100**	99.44	**100**	98.88
Glass	90.65	92.52	88.79	**95.79**	92.52	92.52
Vowel	**100**	**100**	**100**	**100**	**100**	**100**
Vehicle	95.73	97.52	96.10	96.81	95.27	**97.75**
Segment	99.44	99.57	**99.65**	96.81	99.35	99.57
Sat. Image	**92.05**	92.00	91.30	92.00	91.35	91.80
Optical	98.16	98.33	98.66	**98.77**	98.11	97.83
Letter	**97.75**	97.50	97.57	97.50	96.83	97.05

Table 3 Average number of support vectors used per class during testing

Datasets	One against one		One against all		Proposed	
	After norm.	Without norm.	After norm.	Without norm.	After norm.	Without norm.
Iris	67.2	80.2	69.2	70.4	**39.6**	56.1
Wine	183.9	188.0	146.8	281.4	**55.6**	126.0
Glass	341.1	352.4	326.2	423.7	**137.58**	157.93
Vowel	3008.4	3328.8	1262.7	1299.9	**380.37**	452.8
Vehicle	428.8	1375.8	656.5	1608.4	**315.9**	694.7
Segment	1403.1	1971.8	771.5	3838.7	**385.5**	1097.1
Sat. image	2531	7078	3772	7584	**2120.3**	2297.8
Optical	7697	9309	2827	2808	3141.5	**1235.3**
Letter	103,208	123,523	14,493	14,380	**9055.8**	12,317.0

Table 4 Comparing directed binary tree decomposition methods (in %)

Datasets	Proposed method		Method 1	Method 2	Method 3
	After norm.	Without norm.	Centroid distance	Balanced subsets	Scatter measure
Sat. image	91.35	91.80	90.90	91.90	91.80
Optical	98.11	97.83	96.50	96.90	96.20

To compare the proposed approach with other directed binary tree decomposition methods, our results have been compared with those presented in Lorena et al.'s paper [13] (which followed similar experimentation procedure). The comparison is presented in Table 4. Looking at the table, it can be ascertained that the proposed method provides either comparable or superior performance to that of the methods presented in [13]. Further, the added advantage that the proposed approach assures a balanced binary tree-like structure could make it more appealing than its counterparts.

5 Conclusion

The paper presented a brief and succinct review of existing decomposition algorithms for using binary classifiers in multiclass classification problems. The paper then proposed a novel Hausdorff distance-based decomposition algorithm that leads to a balanced binary tree-like partitioning structure. The approach has been elaborately evaluated and compared with OAO and OAA decomposition methods. Inferring from the results, it was observed that the classification performance of the proposed method was comparable to that of OAA and OAO methods. Further, as expected theoretically, testing speed of the proposed method was empirically observed to be faster than OAO and OAA methods for all tested datasets. Proposed method also has the advantage of zero ambiguity during final class assignment due to non-occurrence of shadow region. Future work could be on checking the use of hierarchical clustering methods on classes with Hausdorff distance as the distance measure between them for making the binary tree. Another future work could be on evaluating and comparing the performance of different multiclass algorithms with Fuzzy SVM [16, 20].

References

1. Vapnik, V.: An overview of statistical learning theory. IEEE Trans. Neural Netw. **10**(5) (1999)
2. Sevakula, R.K., Verma, N.K.: Assessing generalization ability of majority vote point classifiers. IEEE Trans. Neural Netw. Learn. Syst. (2016)
3. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2002)
4. Lorena, A.C., Carvalho, A.C.P.L.F., Gama, J.M.: A review on the combination of binary classifiers in multiclass problems. J. Artif. Intell. Rev. **30**(1–4), 19–37 (2008)
5. Allwein, E.L., Shapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. In: Proceedings of the 17th International Conferences on Machine Learning, pp. 9–16 (2000)
6. Dietterich, T.G., Bariki, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. **2**, 263–286 (1995)
7. Pimenta, E., Gama, J.: A study on error correcting output codes. In: Proceedings of the 2005 Portuguese conference on Artificial intelligence (EPIA'05), pp. 218–223 (2005)
8. Platt, J.C., Cristiani, N., Shawe-Taylor, J.: Large margin DAGs for multiclass classification. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 547–553 (2000)

9. Kijsirikul, B., Ussivakul, N.: Multiclass support vector machines using adaptive directed acyclic graph. In: 2002 Proceedings of the International Joint Conference on Neural Networks, pp. 980–985 (2002)
10. Lei, H., Govindaraju, V.: Half-against-half multi-class support vector machines. In: Proceedings of the 6th International workshop on Multiple Classifier Systems, Seaside, CA, pp. 156–164 (2005)
11. Takahashi, F., Abe, S.: Decision-tree-based multiclass support vector machines. In: Proceedings of the 9th International Conferences Neural Information Processing (ICONIP'02), pp. 1418–1422 (2002)
12. Vural, V., Dy, J.G.: A hierarchical method for multi-class support vector machines. In: Proceedings of the 21st International Conferences on Machine Learning, Banff, pp. 831–838 (2004)
13. Lorena, A.C., Carvalho, A.C.P.L.F.: Minimum spanning trees in hierarchical multiclass support vector machines generation. In: International Conferences on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 422–431. Springer (2005)
14. Henrikson, J.: Completeness and total boundedness of the Hausdorff metric. MIT Undergrad. J. Math. **1**, 69–80 (1999)
15. Scharf, L.: Computing the Hausdorff distance between sets of curves. Ph.D. thesis, Freie Universitat, Berlin (2003)
16. Sevakula, R.K., Verma, N.K.: Compounding general purpose membership functions for fuzzy support vector machine under noisy environment. IEEE Trans. Fuzzy Syst. (2017)
17. Chen, J., Wang, C.: Combining support vector machines with a pairwise decision tree. IEEE GeoSci. Remote Sens. Lett. **5**(3), 409–413 (2008)
18. Cortes, C., Vapnik, V.: Support vector networks. Mach. Learn. **20**, 273–297 (1995)
19. Sevakula, R.K., Verma, N.K.: Support Vector Machine for Large Databases as Classifier. Springer SEMCCO 2012, pp. 303–313 (2012)
20. Sevakula, R.K., Verma, N.K.: Clustering based outlier detection in fuzzy SVM. In: 2014 IEEE International Conferences on Fuzzy Systems (FUZZ-IEEE'14), pp. 1172–1177 (2014)
21. Pujol, O., Radeva, P., Vitria, J.: Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes. IEEE Trans. Pattern Anal. Mach. Intell. **28**(6), 1007–1012 (2006)
22. Schwenker, F.: Hierarchical support vector machines for multi-class pattern recognition. In: Proceedings of the Fourth International Conferences on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, pp. 561–565 (2000)
23. Shen, L., Tan, E.C.: Seeking better output-codes with genetic algorithm for multiclass cancer classification (Submitted to Bioinformatics)
24. Lin, C.-F., De Wang, S.: Fuzzy support vector machines. IEEE Trans. Neural Netw. **13**(2), 464–471 (2002)
25. Bordes, A.: Fast kernel classifiers with online and active learning. J. Mach. Learn. 1579–1619 (2005)
26. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Comput. Learn. Theory 35–46 (2000)
27. Liu, B., Hao, Z., Tsang, E.C.C.: Nesting one-against-one algorithm based on SVMs for pattern classification. IEEE Trans. Neural Netw. **19**(12), 2044–2052 (2008)

Part IV
Neuro-Fuzzy Systems

Investigation of RBF Kernelized ANFIS for Fault Diagnosis in Wireless Sensor Networks

Rakesh Ranjan Swain, Tirtharaj Dash and Pabitra Mohan Khilar

Abstract Wireless sensor networks (WSN) are often deployed in human inaccessible environments. Some examples of such environments that are difficult for quick human reach include deep forests, various hazardous industries, hilltops, and sometimes underwater. The occurrence of failures in sensor networks is inevitable due to continuous or instant change in environmental parameters. A failure may lead to faulty readings which in turn may cause economic and physical damages to the environment. In this work, a thorough investigation has been conducted on the application of adaptive neuro-fuzzy inference system (ANFIS) for automated fault diagnosis in WSN. Further, a kernelized version of ANFIS has also been studied for the discussed problem. To avoid the model's undesired biases toward a specific type of failure, oversampling has been done for multiple version of the ANFIS model. This study would serve as a guideline for the community toward the application of fuzzy inference approaches for fault diagnosis in sensor networks. However, the work focuses on the automated fault diagnosis in open air WSN and has no applicability in underwater sensor network systems.

Keywords Fault diagnosis · Wireless sensor networks · Kernelization · ANFIS

R. R. Swain · P. M. Khilar
Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, India
e-mail: rakeshswain89@gmail.com

P. M. Khilar
e-mail: pmkhilar@nitrkl.ac.in

T. Dash (✉)
Department of Computer Science and Information Systems, Birla Institute of Technology and Science Pilani, Goa Campus, Sancoale, India
e-mail: tirtharaj@goa.bits-pilani.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_20

1 Introduction

Wireless sensor network (WSN) is a wireless network constisting of a set of sensory devices (called nodes) that are often automated and are low in cost, power, memory storage. Usually, sensor nodes are small sized and are sometimes multi-functional in nature. A sensor node contains sensory unit, processor, memory storage, actuator, and power supply unit. The sensor nodes are sometimes equipped with multiple sensory units to collect various environmental parameters for sensing environmental conditions. It is well known that WSN has found their applicability in numerous applications such as environmental monitoring, target tracking, health monitoring [1]. Generally, sensor nodes are densely deployed in an unstructured manner for the purpose of sensing, processing, and communicating with each other. In an unstructured deployment in a comparatively human inaccessible environment, failure of sensor nodes is inevitable. A failure in a WSN leads to faulty readings which in turn may cause economic and physical damages to the environment. For example, failures that might occur specifically in human health monitoring environment or in chemical power plants are quite prominent. Diagnosis of faults in WSN is quite a well-studied research for the community [2]. There are four different types of sensor faults that can be seen in WSN such as (1) hard fault, (2) soft permanent fault, (3) intermittent fault, and (4) transient fault. The latter three types of faults can be grouped in the category of soft fault.

Accurate diagnosis of fault in a real-time environment from the sensor readings data is quite a challenge for the community. To the best of our belief, the majority of fault diagnosis works focused on the detection of two broad types of faults such as (a) hard fault and (b) soft fault. These works did not focus more on the three sub-categories of soft faults as mentioned earlier. The present work focuses on the automated diagnosis of the soft faults. Machine learning (ML)-based techniques such as neural networks are very popular approaches in the field of automated fault diagnosis, for example, a work presented in [3, 4], and handling uncertainty by incorporating fuzzy logic [5]. Further, the works carried out in this field have been identified in Sect. 5.

In the present work, we investigate the application of adaptive neuro-fuzzy inference system (ANFIS) for automated fault diagnosis in WSN. Our proposed method has been investigated for all three different types of soft faults. Further, we equip the ANFIS module with radial basis kernel (RBF) to study whether such an approach could improve the performance of conventional ANFIS technique. In real-time WSN, the occurrence of few categories of soft faults is minimal that offers another challenge for an ML approach to accurately diagnose the faults from the available sensor readings. In this work, we also study the automatic synthesis of faulty sensor readings from the available data using a relative distance-based oversampling technique such as SMOTE.

The rest of the paper is presented as followed. We present our proposed kernelized ANFIS approach in Sect. 2. The experimental design has been studied in Sect. 3. Results obtained by the proposed approach have been discussed in Sect. 4. The work has been concluded in Sect. 6.

2 Kernelized ANFIS

ANFIS is a special class of multilayer adaptive networks that incorporate both neural networks (NN) and fuzzy logic principles to model uncertain systems [6] such as WSN. Neural networks are supervised learning models that approximate the underlying transformation function from the historical input–output data pairs. The function suggested by the NN could further be used for future predictions where the output is unknown. In fuzzy logic, a rule base drawn from the available historical data and the control signal is generated by firing the rule base. ANFIS makes the selection of the rule base more adaptive to the situation and the problem under consideration.

A typical architecture of ANFIS has been shown in Fig. 1. The architecture shown in the figure consists of two inputs, four rules, and one output for the first-order Sugeno fuzzy model. Here each input is assumed to have two associated membership functions (MFs). So there are four possible IF-THEN rules. For a first-order Sugeno-type fuzzy model, a typical rule can be expressed as

$$\text{If } x \text{ is } A_i \text{ and } y \text{ is } B_j, \text{ then } f_{ij} = a_{ij}x + b_{ij}y + c_{ij} \tag{1}$$

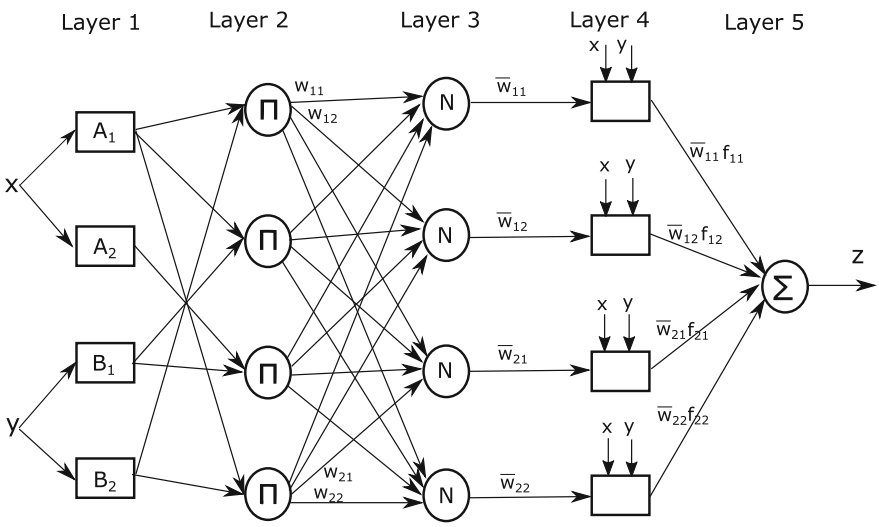


Fig. 1 ANFIS architecture

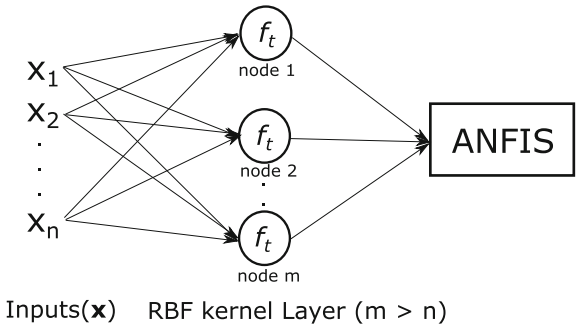


Fig. 2 Kernelized ANFIS architecture (the ANFIS block this figure uses the layer 1 through to layer 5 of Fig. 1 where the inputs to the ANFIS are kernelized using radial basis functions)

where A_i and B_j are the membership functions (MF) for the inputs x and y, respectively. The constants a_{ij}, b_{ij}, and c_{ij} are consequent parameters [6]. A set of such rules can be obtained by changing the inputs or the MF. The functionality of each layer can be briefed as follows.[1] The layer 1 is an adaptive layer that generates the membership grades for the inputs using the MF. The MF parameters in this layer are called the premise or antecedent parameters. The layer 2 computes the firing strength of each of the rules by doing simple multiplication or minimum operation, usually denoted as Π. Layer 3 computes the normalized firing strength by taking individual firing strengths from the layer 2. Each node in the layer 4 is an adaptive node. The output of a node in layer 4 is the product of the normalized firing strength and the first-order polynomial of the first-order Sugeno model. The last layer (layer 5) computes the final output by taking the summation of the outputs from the previous layer. Learning of the parameters in layer 1 (premise parameters) and layer 4 (consequent parameters) happens by the squared error signal generated at the output and with the help of a suitable optimization algorithm such as conjugate gradient method.

The hypothesis that we are testing in this work is that a problem, when cast in higher dimensional feature space is more likely to be separable than that in a lower dimensional space [7]. The present work investigates this hypothesis by adding a nonlinear transformation layer where the original inputs are transformed to a new feature space in a higher dimension. Each computational unit j in this layer does a nonlinear transformation (f_t in Fig. 2) using RBF kernel defined in Eq. (2) of the input data pattern which is denoted as $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

$$f_t = \varphi(\mathbf{x}) = \varphi(\mathbf{x} - \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma_j^2}||\mathbf{x} - \mathbf{x}_j||^2\right), j = 1, 2, \ldots, m \qquad (2)$$

In Eq. (2), $\mathbf{x}_j$ is the center of the data points; $||\cdot||$ represents the Euclidean distance. The number of neurons in the RBF transformation layer m has been decided by applying the k-means clustering [8] on the available training data with only the input

[1]For page constraints, we have intentionally omitted major mathematical background behind the functionality of each layer. Readers are advised to go through the main ANFIS article by Professor Jang [6].

features. It is believed that patterns sharing common characteristics would be grouped together. If there are N patterns in a dataset, we decided that the value of k in k-means clustering is strictly lesser than N ($k < N$), and strictly higher than the number of features of the data, that is $k > n$ to support our initial hypothesis of transformation to higher dimension. It should be noted that $m = k$.

In reality, sensor data from the faulty sensors are available in few chunks than the normal data. These sensors might belong to any of the three classes of the soft faults. We employed synthetic minority oversampling technique (SMOTE) [9] to generate the sensor data that are similar to the available data. This approach also adds an investigation of the synthetic sampling technique to the fault diagnosis in WSN. The final algorithm has been presented in the following.

Data: Sensored data from the central receiver
Result: Fault diagnosis results on unseen test data
Step 0: Initialization and setting necessary ANFIS parameters and m;
Step 1: Partition the available data into training data (D_{train}) and testing data (D_{test}) by random sampling of the patterns from the whole dataset;
Step 2: Obtain the oversampled data, $D_{osTrain}$ = SMOTE(D_{train});
Step 3: Apply k-means algorithm on $D_{osTrain}$ without the class labels and obtain centroids of each clusters;
Step 4: Apply RBF kernelization scheme to get the data ($D_{rbfOsTrain}$) for ANFIS training;
Step 5: Train ANFIS on $D_{rbfOsTrain}$ using conjugate gradient method;
Step 6: Test the trained ANFIS using the (D_{test});

Algorithm 1: RBF kernelized ANFIS with oversampling for fault diagnosis in WSN

3 Design of Experiment

The present work is a simulation-based experiment where the data are obtained from real-world experimental setup as explained further.

The experiment was conducted in an outdoor environment. In this experiment, 10 sensor nodes were deployed in an area of $5 \times 5\,\text{m}^2$. Figure 3 shows the deployment of all these sensor nodes. Each sensor module is designed by Arduino Uno embedded and At mega 8-bit microcontroller, DHT11 temperature sensor, MRF24J40MA transceiver, and 2200 mAh battery. All these sensor nodes are communicated to the base station with a gateway node. The diagnosis program is run in the base station. All the sensor nodes and the gateway node are presented in the transmission range of each other. The transmission power is set -16 dBm for the communication range in between 5 and 6 m. The gateway node is presented in between 6 m distance.

Initially, all the nodes were considered fault free in nature. The sensor node temperature data were collected every hour of a daytime. Then after analyzed the normal sensor data, we set the minimum threshold temperature value θ_1 and maximum

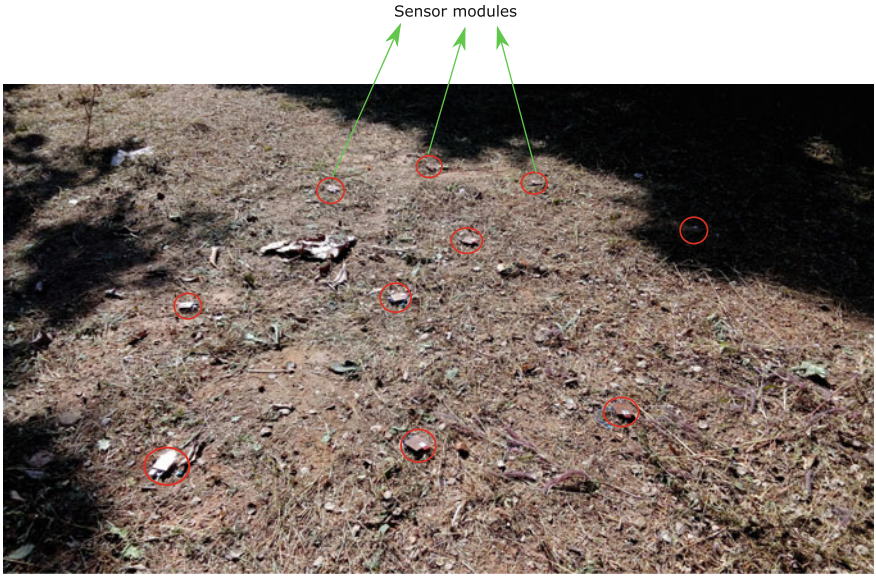


Fig. 3 Experimental setup in an outdoor environment. A gateway node communicates with these sensor modules

threshold value θ_2. The sensor node data within the θ_1 and θ_2 are considered as fault free and exceeding the range, it is considered as faulty. Let x_i be the sensor data from sensor node n_i at tth time instance. So, the sensor data are denoted as $x_i(t)$. For sensor node n_i, we collect the sensor data $\mathbf{x} = (x_i(t))$; $t = 1, \ldots, T$. From the time $t = 1$ to T, all the sensor data were analyzed. The data $x_i(t)$ follow the normal distribution $\mathcal{N}(\mathbf{x}_i; \sigma_i^2)$, where $\mathbf{x}_i$ is the actual data and σ_i^2 is the variance of fault data at n_i sensor node (noise to make the node faulty). So this value, we are represented as in Eq. (3). Similarly, depending on the probability of noisy data, the faulty sensor readings could be generated for different types of faults.

$$\mathbf{x}_i(t) = \mathbf{x}_i + e_i(t); \;\; e_i \sim \mathcal{N}(\mathbf{x}_i; \sigma_i^2) \tag{3}$$

All the ANFIS-based simulations have been conducted in MATLAB R2017a on a quad-core system with 4 GB RAM. After obtaining the sensor readings (including the faulty sensor readings data), the whole dataset was partitioned for the training and testing simulation of various ANFIS approaches. The train and test partitioning were done by randomly selecting the patterns as per 70%: 30% ratio, respectively. For all the ANFIS experiments, the number of training epochs was set to 100. The dataset summary has been reported in Table 1.

It should be noted that the oversampling was only applied to the training dataset. The test dataset remains intact throughout the experiment except for the RBF kernelization experiment in which both the training and testing are to be transformed to

Table 1 Available sensor data summary (N: total number of patterns, $N(class = i)$: number of patterns of ith class)

Data	N	$N(class = 1)$	$N(class = 2)$	$N(class = 3)$	$N(class = 4)$
Original	5000	2000	500	1500	1000
Train (D_{train})	3500	1398	333	1064	705
Test (D_{test})	1500	602	152	452	294

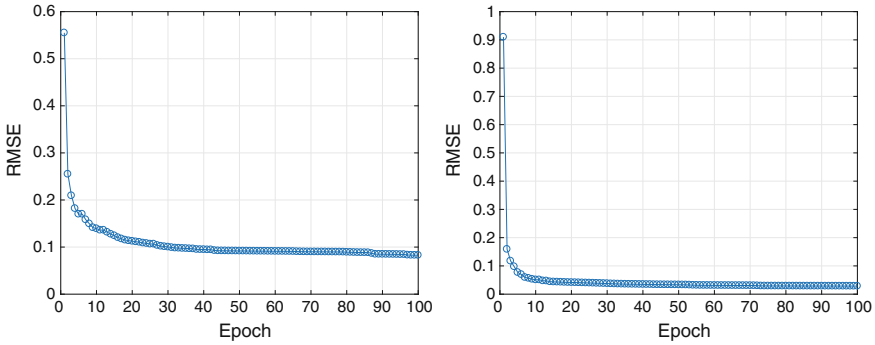
a new higher dimensional feature space where the centroids were obtained using the k-means clustering on the training data only. Further, the results are analyzed in the following section.

4 Results

The spread of the RBF kernel σ plays an important role as an amplifier of the distance between a pattern and the center of the kernel. If this distance is much larger than σ, the kernel function tends to be very minimal. Moreover, it has a high impact on the decision rules—specifically, boundary—that can affect the prediction of the testing data. Therefore, the results for two different values of the spread are noted (a) $\sigma = 5$, (b) $\sigma = 10$. The number of subgrouping(clustering) in the available training data m has been varied from the 15 to 25 with equal separation of 5 to investigate the effect of the transformed feature space in a slightly higher dimension ($m \approx n$; $m = 15, n = 10$) and a dimension as large as twice the input feature space ($m > 2n$; $m = 25, n = 10$).

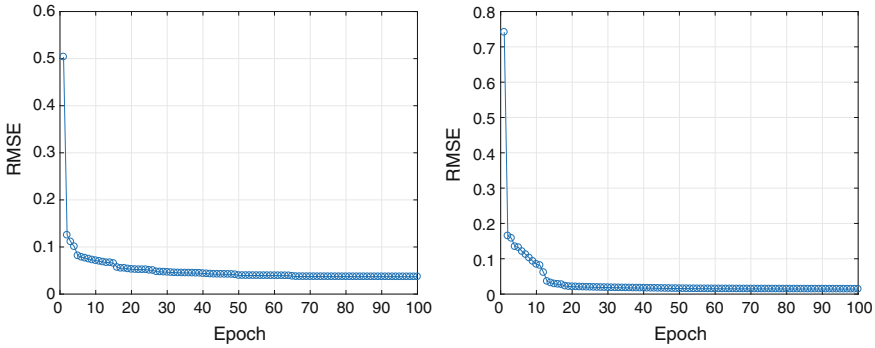
We used root mean squared error (RMSE) as a performance evaluation measure of ANFIS models. A few sample training error convergence curves are presented in Figs. 4, 5, and 6. For the further investigate the quality of fault diagnosis in terms of the faulty nodes prediction rate, we used the accuracy measure. We also report the prediction rate after training and after testing to clearly investigate the specialization and generalization capability of the developed ANFIS models. These results are summarized in Table 2 where m is the size of the transformed feature space and σ is the spread of the RBF kernel.

The present investigation suggests that a higher dimensional transformation of the input feature space prior to the learning rules by the ANFIS might not lead to an improved predictive capability of the models for the present problem on the faulty diagnosis from the available data. It can be revealed from the data reported in Table 2. The conventional ANFIS and the ANFIS with SMOTE oversampling are quite comparable with each other based on their presently achieved generalization capability on the test dataset. However, the RBF kernelized ANFIS with and without oversampling are very highly specialized toward the training data that is leading to over-fitting and in turn poor generalization. Hence, such a model might not be



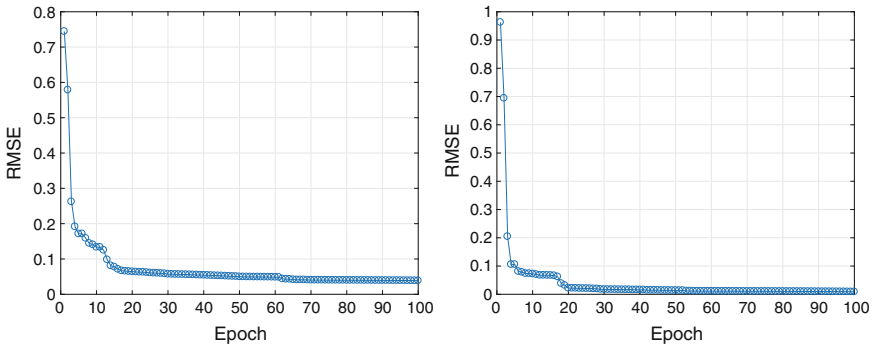
(a) ANFIS (b) ANFIS-SMOTE

Fig. 4 Error convergence during training



(a) ANFIS $(m = 15, \sigma = 5)$ (b) ANFIS $(m = 15, \sigma = 5)$-SMOTE

Fig. 5 Error convergence during training



(a) ANFIS $(m = 25, \sigma = 10)$ (b) ANFIS $(m = 25, \sigma = 10)$-SMOTE

Fig. 6 Error convergence during training

Table 2 Fault diagnosis (accuracy of prediction) scores achieved by various ANFIS approaches (RMSE has been noted for Epoch 1 and the Epoch 100 shown as transition '→')

Method	Training (%)	Testing (%)	RMSE
ANFIS	95.88	94.80	0.5549 → 0.0827
ANFIS-SMOTE	98.55	94.20	0.9098 → 0.0285
ANFIS ($m = 15, \sigma = 5$)	98.17	16.46	0.6201 → 0.0369
ANFIS ($m = 15, \sigma = 10$)	93.60	16.80	0.7435 → 0.1135
ANFIS ($m = 20, \sigma = 5$)	79.82	34.53	0.5874 → 0.4002
ANFIS ($m = 20, \sigma = 10$)	88.51	20.20	0.7795 → 0.2025
ANFIS ($m = 25, \sigma = 5$)	99.14	32.60	0.5292 → 0.0179
ANFIS ($m = 25, \sigma = 10$)	97.88	29.86	0.7442 → 0.0381
ANFIS ($m = 15, \sigma = 5$)-SMOTE	99.28	17.93	0.7415 → 0.0141
ANFIS ($m = 15, \sigma = 10$)-SMOTE	99.47	18.53	0.9703 → 0.0099
ANFIS ($m = 20, \sigma = 5$)-SMOTE	99.60	38.73	0.9095 → 0.0076
ANFIS ($m = 20, \sigma = 10$)-SMOTE	99.33	22.26	1.0170 → 0.0111
ANFIS ($m = 25, \sigma = 5$)-SMOTE	99.85	35.00	0.7919 → 0.0029
ANFIS ($m = 25, \sigma = 10$)-SMOTE	99.51	30.26	0.9625 → 0.0089

preferable for the present problem. However, further experiments must be conducted carefully before concluding in a general sense.

5 Related Works

Several different soft computing or statistical techniques have been employed recently for studying the fault diagnosis in WSN. A statistical approach called neigh-

boring coordination method was used where the authors used the statistical variance of the sensor readings for predicting the fault [10]. A Takagi–Sugeno–Kang (TSK)-based fuzzy inference system was applied for detection of soft permanent faults and transient faults in WSN [11]. Fuzzy transformation-based multilayer feed-forward neural network [12] has been successfully implemented for automated fault diagnosis in WSN [13]. However, this work used a single membership function to generate a single transformed feature instead of three different features as in our present work. There are many different approaches to automated fault diagnosis in WSN that use different kinds of neural networks such as probabilistic neural network [14] and statistical techniques such as hypothesis testing [10]. Many different kinds of neural networks have been developed recently for use in fault diagnosis and could be referred to [15, 16].

6 Conclusion

Many different research works present the success stories behind the application of various methods to automated fault diagnosis in WSN. The present work focused on a thorough investigation of the possible modification to fuzzy inference-based investigation scheme using kernelized feature space. The achieved results suggested that the application of such a scheme might not be suitable for the problem under consideration because of the fact that it makes the inference engine highly specialized toward the training distribution and makes the machine weak in generalization which is the most important aspect of learning machines. On the basis of the obtained results, one could further study in the following possible directions.

Our primary results show that ANFIS with SMOTE without the RBF kernelization outperforms ANFIS during training. However, it has identical results in testing. This leaves us with a question to investigate while using other learning machines. Specifically, we should be interested in using oversampling for rule generators such as decision tree and random forest classifiers. It could be hypothesized that an oversampling engine would generate missing information for the sensor dataset and could make it complete. Then a decision tree can be used to learn the set of rules for the input–output transformation. Further, such a model is not computationally expensive unlike sophisticated models like neural networks.

The results achieved due to the kernel trick in the ANFIS (and ANFIS with SMOTE) is worse than its non-kernelized counterpart because of poor generalization of the underlying learning machine. A straightforward question to investigate in future is to study the use of support vector machines (SVM) with kernel trick for automated classification of sensor faults. As discussed in the aforementioned point, such a method would also be computationally inexpensive as compared with various neural networks.

Furthermore, failures in sensor networks might occur in both sensor nodes and the links. It might be arguably correct to say that failures in linked sensor nodes would cause link failure. But, for instance, the primary focus is that such failures lead to

the formation of islands or cuts in sensor networks. Earlier studies have successfully used simple graph theoretic approaches such as [17] for detection of cuts in WSN. Sometimes third-party intrusion might cause such failures in the network. So, in such circumstances, a robust model must be designed for automated diagnosis of failures that would not only detect failures but also decide whether the failure was caused due to environmental damages or third-party intrusion [18–20]. To the best of our knowledge, such a model that would deal with both these kinds of decisions has not been built.

References

1. Arampatzis, T., Lygeros, J., Manesis, S.: A survey of applications of wireless sensors and wireless sensor networks. In: Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, pp. 719–724. IEEE (2005)
2. Mahapatro, A., Khilar, P.M.: Fault diagnosis in wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. **15**(4), 2000–2026 (2013)
3. Hou, L., Bergmann, N.W.: Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis. IEEE Trans. Instrum. Meas. **61**(10), 2787–2798 (2012)
4. Swain, R.R., Khilar, P.M.: Composite fault diagnosis in wireless sensor networks using neural networks. Wirel. Pers. Commun. **95**(3), 2507–2548 (2017)
5. Reddy, P.N., Dambekodi, S.N., Dash, T.: Towards continuous monitoring of environment under uncertainty: a fuzzy granular decision tree approach. In: CEUR Workshop Proceedings, vol. 1819 (2017)
6. Jang, J.S.: Anfis: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**(3), 665–685 (1993)
7. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans. Electron. Comput. **3**, 326–334 (1965)
8. Hartigan, J.A., Wong, M.A.: Algorithm as 136: a k-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
10. Panda, M., Khilar, P.M.: Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test. Ad Hoc Netw. **25**, 170–184 (2015)
11. Khan, S.A., Daachi, B., Djouani, K.: Application of fuzzy inference systems to detection of faults in wireless sensor networks. Neurocomputing **94**, 111–120 (2012)
12. Dash, T., Behera, H.S.: A fuzzy MLP approach for non-linear pattern classification (2015). arXiv:1601.03481
13. Swain, R.R., Khilar, P.M.: A fuzzy MLP approach for fault diagnosis in wireless sensor networks. In: 2016 IEEE Region 10 Conference (TENCON), pp. 3183–3188. IEEE (2016)
14. Swain, R.R., Khilar, P.M., Bhoi, S.K.: Heterogeneous fault diagnosis for wireless sensor networks. Ad Hoc Netw. **69**, 15–37 (2018)
15. Ji, Z., Bing-shu, W., Yong-guang, M., Rong-hua, Z., Jian, D.: Fault diagnosis of sensor network using information fusion defined on different reference sets. In: 2006 International Conference on Radar, CIE'06, pp. 1–5. IEEE (2006)
16. Moustapha, A.I., Selmic, R.R.: Wireless sensor network modeling using modified recurrent neural networks: application to fault detection. IEEE Trans. Instrum. Meas. **57**(5), 981–988 (2008)
17. Swain, R.R., Dash, T., Khilar, P.M.: An effective graph-theoretic approach towards simultaneous detection of fault(s) and cut(s) in wireless sensor networks. Int. J. Commun. Syst. (2017)

18. da Silva, A.P.R., Martins, M.H., Rocha, B.P., Loureiro, A.A., Ruiz, L.B., Wong, H.C.: Decentralized intrusion detection in wireless sensor networks. In: Proceedings of the 1st ACM International Workshop on Quality of Service and Security in Wireless and Mobile Networks, pp. 16–23. ACM (2005)
19. Roman, R., Zhou, J., Lopez, J.: Applying intrusion detection systems to wireless sensor networks. In: IEEE Consumer Communications and Networking Conference (CCNC'06) (2006)
20. Dash, T.: A study on intrusion detection using neural networks trained with evolutionary algorithms. Soft Comput. **21**(10), 2687–2700 (2017)

On Adaptive Control for AGV with Vision Sensor as an Unknown Nonlinear System

Pankaj Kumar Mishra, Pooja Agrawal, Narendra Kumar Dhar and Nishchal K. Verma

Abstract Adaptive control using neural network provides a real-time systematic approach to achieve or maintain a desired level of control system performance when dynamics are unknown. The paper proposes a novel approach for designing of an adaptive controller with input as relative pixel density from a fuzzy system for automated guided vehicle with vision sensor. The fuzzy system computes relative pixel density from vision sensor data while minimizing uncertainties due to illumination, occlusion, and obscure images. It provides a methodology to apply an advanced nonlinear intelligent control technique for vision-based path tracking problem. The proposed strategy has been applied for path tracking problem on an indigenously developed vehicle. The results obtained show the efficiency of proposed approach and ease of applying different control techniques for vision sensor-based plant.

Keywords Adaptive controller · Fuzzy system · Neural network relative pixel density · Path tracking · Automated guided vehicle · AGV

1 Introduction

Automatic guided vehicles (AGVs) are load carriers or material handler that travel along the guided path or predictable path for transportation of stuff, work in process, and finished products. With the development of industry, it has been widely used

P. K. Mishra (✉) · P. Agrawal · N. K. Dhar · N. K. Verma
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: pkmishra@iitk.ac.in

P. Agrawal
e-mail: pooja.iitd@gmail.com

N. K. Dhar
e-mail: nardhar@iitk.ac.in

N. K. Verma
e-mail: nishchal@iitk.ac.in; nishchal.iitk@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_21

where high volume of repetitive movements of material is required or no human higher cognitive process ability is needed to perform the movement. In traditional mode of transportation and production in industry, most of the tasks are done by the human or by the manually driven equipment where transport, handling, and storage take 90% of the total time [1, 2].

AGV with different types of sensors which can work in high and low temperature, high noise and smell of gas is a good solution to attain flexibility, automation, intelligent production workshop with a contemporary supply system [3]. The modern industrial system generally uses laser navigation or inertial navigation technique to follow a predefined path which requires costly sensor. However, the recent advances in the field of computer vision have made possible of integration of the camera with AGV. But, due to the vision complex algorithm and unavailability of physical dynamics of the AGV it has been a challenging task for real-time system. Traditional vision-based control scheme involving the calculation of geometric relationships between the received image and target image was based on homography estimation and epipolar geometry [4–7]. In [8–11], image-based schemes are used for vision-based control, which requires omnidirectional camera, image-based memory and involves Jacobian estimation and inverse transformation. Some surveys on visual servo control are [12, 13]. In recent years, a number of works [14–20] have been done on vision based path tracking scheme. All these approaches lack the methodology to apply an advanced intelligent control scheme which makes them less robust and inefficient in handling uncertainty.

In this paper, a novel idea of using AGV and camera as an unknown nonlinear system is applied to design a controller for path tracking. Relative change in pixel density along the tracking path is considered as a state of the system. This approach has broken down a computer vision problem as a controller design problem for an unknown nonlinear system and also made an open area for the implication of an advanced control system technique. The controller is designed as [21], where actions are based on the data from a single camera placed on the automated vehicle. The data is sent to a remotely placed controller, which generates required control actions and sends them to actuators. The data transfer takes place through a wireless medium based on a communication framework [22].

The paper is organized as follows. The design of the robust adaptive neural network controller is discussed in next section. Section 3 contains results and discussions of the real-time tracking operation performed by the vehicle. Finally, the paper concludes the entire work.

2 Controller Design

The automated vehicle has an RGB camera which continuously monitors the desired path and provides data to controller. Our system design is directed toward control of model-free vehicle. Hence, the details of automated vehicle such as dynamic modeling and system equations are ignored.

The image data is operated by a fuzzy system before passing it as an input to adaptive controller. The computation of vehicle's relative position using fuzzy approach is explained in the following subsection.

2.1 Computing Relative Pixel Density Using Fuzzy

Camera as a vision sensor is the only mode for data acquisition and feedback during real-time operation. The vision data is obtained from precision camera fixed on the top-front part of automated vehicle. The image frames are captured continuously, while vehicle is in motion along the desired track. Each frame is processed by dividing it into several regions depending on the tracking precision. Figure 1 shows several regions of image frame. Pixels in different regions are assigned different values. Figure 2 shows the schematic of system architecture with automated vehicle.

Image data obtained in real time may have uncertainties due to illumination effects or noises. Fuzzy system is used for decision making in terms of relative position of vehicle with respect to path. Fuzzy system tries to implicate nonlinear behavior of path, which otherwise looks difficult to interpret or analyze. The membership

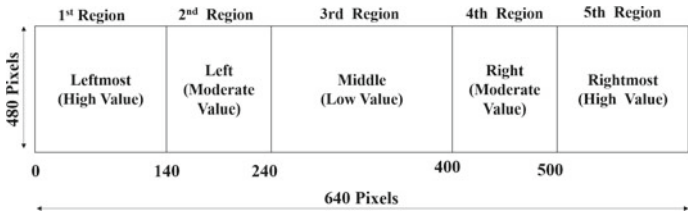


Fig. 1 A sample image segmented into different regions

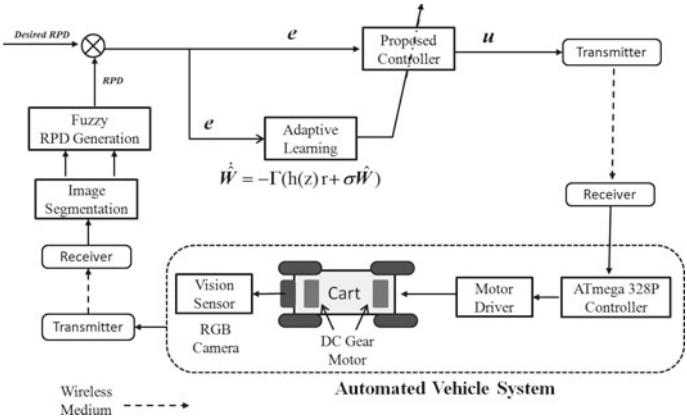


Fig. 2 Schematic of system architecture with automated vehicle

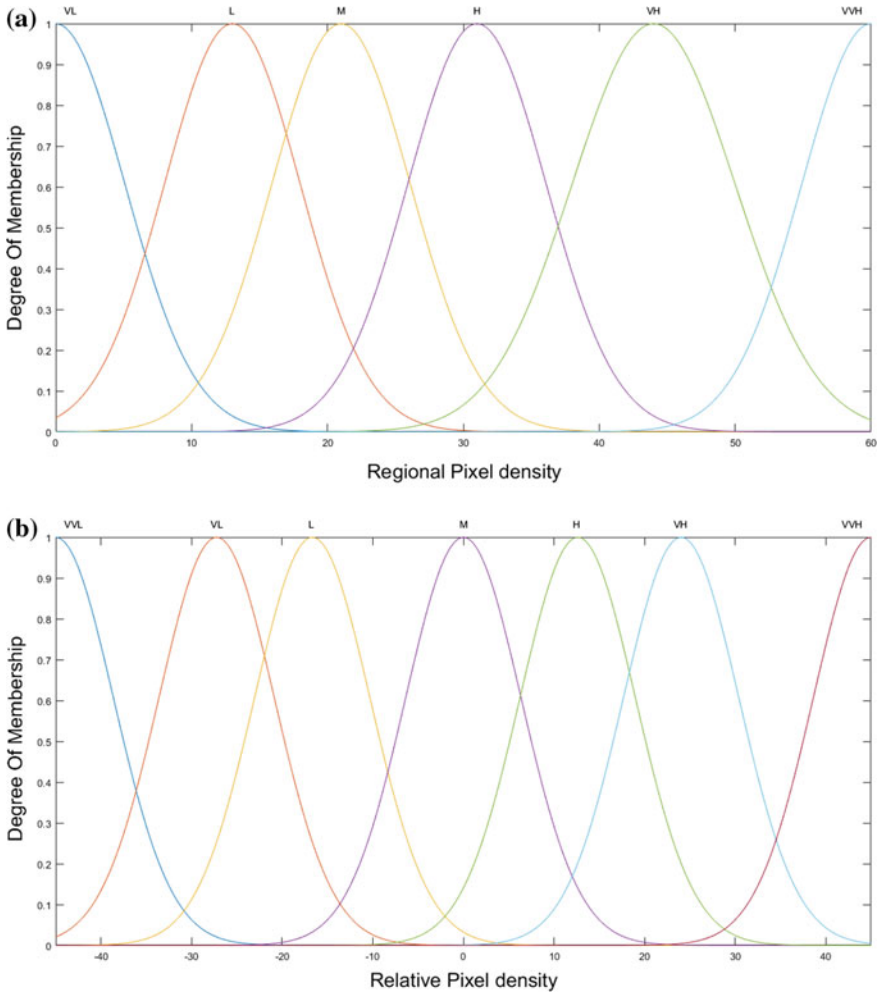


Fig. 3 Fuzzy membership functions for **a** input and **b** output

functions of fuzzy system are learnt using genetic algorithm (GA) [23]. The pixel density (PD) of segmented regions acts as input for the fuzzy system. The image frame is divided into two sections at the middle, i.e., left and right section. Relative pixel density (RPD) for current frame is computed by interval fuzzy system.

The universe of discourse for pixel densities PD_{left} and PD_{right} is divided into six fuzzy regions as shown in Fig. 3a, while the output RPD is divided into seven regions as shown in Fig. 3b. Centroid of area (COA) defuzzification method is used to obtain crisp values of RPD.

Figure 4 shows the rule base designed from vehicle's past behavior and human supervisory data.

Regional PD_{Left}

Regional PD_{Right}

	VL	L	M	H	VH	VVH
VL	M	L	L	VL	VVL	VVL
L	H	M	L	L	VL	VL
M	H	H	M	L	L	VL
H	VH	H	H	M	L	L
VH	VVH	VH	H	H	M	L
VVH	VVH	VH	VH	H	H	M

VVL- Very Very Low
VVH- Very Very High
VL- Very Low
VH- Very High
L- Low
M- Medium
H- High

Fig. 4 Rule base for path tracking by automated vehicle

2.2 Control Input Design

The pulse width modulation (PWM) values for motor (for speed control of wheels) are not linearly dependent on error. The nonlinearity can be modeled accurately by using Gaussian membership function. The membership function parameters, i.e., mean and variance, are learnt using GA with fitness function as $error = 0.5\sum(PWM_{error})^2$.

The output from fuzzy system is used as an input to the controller. The controller design involves function approximation as system dynamics is not known. The controller follows a direct neural network architecture.

Notation Let $\|(x)\| = (x^t x)^{1/2}$ and $\|\mathbf{X}\| = (trace(X^T X))^{1/2}$ be Euclidean norm of a vector and Frobenius norm of a matrix, respectively. If $x(t) \in L_\infty$, then the norm of the signal is defined as $\|x\| = \text{ess}\ \underset{t}{sup}(|x(t)| < \infty)$.

System Description Considering a SISO nonlinear system in the following form:

$$\dot{\zeta}_1 = \zeta_2; \quad \dot{\zeta}_2 = f(\zeta) + g(\zeta)u + d(t); \qquad y = \zeta_1 \tag{1}$$

where $\zeta = [\zeta_1 \quad \zeta_2]^T \in \mathbb{R}^2$ is the state vector, ζ_1 and ζ_2 are relative pixel density around the path and its rate of change respectively, $f(\zeta)$ and $g(\zeta)$ are unknown smooth nonlinear functions, $u \in \mathbb{R}$ is a plant input, and $d(t)$ is an external disturbances satisfying $|d(t)| \leq d_0$ and $d_0 > 0$.

Assumption 1 There exists a known positive constant $\bar{g}$ such that $0 < |g(\zeta)| \leq \bar{g}$.

Define vector ζ_d as desired RPD and $e = \zeta_1 - \zeta_d$ as error between actual and desired RPD.

Considering tracking error r as

$$r = \lambda e + \dot{e} \tag{2}$$

where λ is constant such that the root of the polynomial related to $r = 0$ is Hurwitz. Using Eqs. (1) and (2)

$$\begin{aligned}
\dot{r} &= \lambda\dot{e} + \ddot{e} = \lambda\dot{e} + \ddot{\zeta}_1 - \ddot{\zeta}_d = \lambda\dot{e} + \dot{\zeta}_2 - \ddot{\zeta}_d \\
\Rightarrow \quad \dot{r} &= \lambda\dot{e} + f(\zeta) + g(\zeta)u + d(t) - \ddot{\zeta}_d \\
\Rightarrow \quad \dot{r} &= f(\zeta) + g(\zeta)u + v + d(t)
\end{aligned} \tag{3}$$

where $v = \lambda\ddot{e} - \ddot{\zeta}_d$.

Desired Control Input and Function Approximation Assuming system (1) to satisfy Assumption 1 and $d(t) = 0$, the desired controller is designed as:

$$u^* = -\frac{1}{g(\zeta)}(f(\zeta) + v) - \left(\frac{1}{\varepsilon g(\zeta)} + \frac{1}{\varepsilon g^2(\zeta)} - \frac{\dot{g}(\zeta)}{2g^2(\zeta)}\right) r \tag{4}$$

where ε is a design parameter satisfying $\varepsilon > 0$.

Using Lyapunov stability theory, it can be shown that the tracking error r is bounded. The proof is given below:

Proof Using (3), (4), and $d(t) = 0$

$$\dot{r} = -\left[\frac{1}{\varepsilon} + \frac{1}{\varepsilon g(\zeta)}\right] r + \frac{\dot{g}(\zeta)}{2g(\zeta)} r \tag{5}$$

Choosing Lyapunov function candidate $V_d = r^2/2g(\zeta)$ and differentiating it, and using (5) and Assumption 1

$$\begin{aligned}
\dot{V}_d &= \frac{1}{g(\zeta)} r\dot{r} - \frac{\dot{g(\zeta)}}{2g^2(\zeta)} r^2 = -\frac{r^2}{\varepsilon g(\zeta)} - \frac{r^2}{\varepsilon g^2(\zeta)} \\
\Rightarrow \quad \dot{V}_d &= -\left(\frac{1}{\varepsilon g(\zeta)} + \frac{1}{\varepsilon g^2(\zeta)}\right) r^2 \leq 0
\end{aligned} \tag{6}$$

Since $\dot{V}_d \leq 0$, which implies $r \in L_\infty$. Subsequently, $e \in L_\infty$.

Remark 1 Equation (6) infers that the parameter ε can be used as a tuning parameter for the converging rate of tracking error.

Since nonlinear functions $f(\zeta)$ and $g(\zeta)$ are unknown, u is not available. So, radial basis function (RBF) architecture of neural network is used for approximation of control action $u^*(\mathbf{z})$.

Where $\mathbf{z} = \left[\zeta, r, \frac{r}{\varepsilon}, v\right]^T \in \mathbf{\Omega}_z \subset \mathbb{R}^5$ and $\mathbf{\Omega}_z$ is a compact set.

Let us consider

$$u^*(\mathbf{z}) = \mathbf{W}^{*T} h(\mathbf{z}) + \mu, \forall \mathbf{z} \in \mathbf{\Omega}_z \subset \mathbb{R}^5 \tag{7}$$

where $h(\mathbf{z})$ is RBF vector, μ is neural network (NN) approximation error satisfying $|\mu| \leq \mu_0$, and $\mathbf{W}^*$ is an unknown ideal constant NN weight vector such that:

$$\mathbf{W}^* := arg \quad \min_{\mathbf{W}} \left\{ \sup_{z \in \Omega_z} |\mathbf{W}^T h(\mathbf{z}) - u^*(\mathbf{z})| \right\} \tag{8}$$

Since for the design of desired control action, ideal weight $\mathbf{W}^*$ is unknown so an adaptive law must be designed for the NN weight vector to achieve the control objective.

Remark 2 ε and $\frac{r}{\varepsilon}$ are dependent. A very small ε chosen for fast tracking error convergence can change their scales. So, $\frac{r}{\varepsilon}$ is also fed into NN as an input for improving approximation.

Adaptive Law Design and Performance Analysis Let NN-based control input be:

$$u = \hat{\mathbf{W}}^T h(\mathbf{z}) \tag{9}$$

and adaptive law be:

$$\dot{\hat{\mathbf{W}}} = -\mathbf{\Gamma}\left(h(\mathbf{z})r + \sigma\hat{\mathbf{W}}\right) \tag{10}$$

where $\hat{\mathbf{W}}$ is an estimate of $\mathbf{W}^*$, $\mathbf{\Gamma} = \mathbf{\Gamma}^T = \gamma I_{l\times l}$ is an adaptive gain matrix, l is number of hidden neurons, and $\sigma > 0$ is a constant parameter.

Substituting u from (9) in (3), $\dot{r}$ can be written as:

$$\dot{r} = f(\zeta) + v + g(\zeta)\hat{\mathbf{W}}^T h(\mathbf{z}) + d(t) \tag{11}$$

Adding and subtracting u^* from (7) on RHS of (11), we have:

$$\begin{aligned}\dot{r} = {} & f(\zeta) + v + g(\zeta)\left(\hat{\mathbf{W}}^T h(\mathbf{z}) - \mathbf{W}^{*T} h(z) - \mu\right) \\ & + g(\zeta)u^*(\mathbf{z}) + d(t)\end{aligned}$$

Substituting u^* from (4) in above equation leads to:

$$\dot{r} = g(\zeta)(\tilde{\mathbf{W}}^T h(\mathbf{z}) - \mu) - \left(\frac{1}{\varepsilon} + \frac{1}{\varepsilon g(\zeta)} - \frac{\dot{g}(\zeta)}{2g(\zeta)}\right) r + d(t) \tag{12}$$

where $\tilde{\mathbf{W}} = \hat{\mathbf{W}} - \mathbf{W}^*$. For NN-based controller performance and design analysis, the Lyapunov function is chosen as:

$$L = \frac{1}{2}\left[\frac{r^2}{g(\zeta)} + \tilde{\mathbf{W}}^T \mathbf{\Gamma}^{-1}\tilde{\mathbf{W}}\right] \tag{13}$$

Remark 3 To avoid $g(\zeta)$ in the adaptive law, $\frac{1}{2}\frac{r^2}{g(\zeta)}$ is used as a part of Lyapunov function instead of $\frac{1}{2}r^2$.

Differentiating (13) and using (10) and (12), we have:

$$\begin{aligned}\dot{L} &= \frac{r\dot{r}}{g(\zeta)} - \frac{r^2 \dot{g(\zeta)}}{2g^2(\zeta)} + \tilde{\mathbf{W}}^T \mathbf{\Gamma}^{-1} \dot{\hat{\mathbf{W}}} \\ &= \frac{r}{g(\zeta)} \left[g(\zeta)(\tilde{\mathbf{W}}^T h(\mathbf{z}) - \mu) - \left(\frac{1}{\varepsilon} + \frac{1}{\varepsilon g(\zeta)} - \frac{\dot{g(\zeta)}}{2g(\zeta)} \right) r \right] \\ &\quad + \frac{d(t)}{g(\zeta)} r - \frac{\dot{g}(\zeta)}{2g^2(\zeta)} r^2 + \tilde{\mathbf{W}}^T \mathbf{\Gamma}^{-1} (-\mathbf{\Gamma}(h(\mathbf{z})r + \sigma \hat{\mathbf{W}})) \\ &= -\left(\frac{1}{\varepsilon g(\zeta)} + \frac{1}{\varepsilon g^2(\zeta)} \right) r^2 + \frac{d(t)}{g(\zeta)} r - \mu r - \sigma \tilde{\mathbf{W}}^T \hat{\mathbf{W}}\end{aligned}$$

Using general relations:

$$\begin{aligned}2\tilde{\mathbf{W}}^T \hat{\mathbf{W}} &= \tilde{\mathbf{W}}^T (\tilde{\mathbf{W}} + \mathbf{W}^*) + (\tilde{\mathbf{W}} - \mathbf{W}^*)^T \hat{\mathbf{W}} \\ &= \left\| \tilde{\mathbf{W}} \right\|^2 + \left\| \hat{\mathbf{W}} \right\|^2 - \| \mathbf{W}^* \|^2 \geq \left\| \tilde{\mathbf{W}} \right\|^2 - \| \mathbf{W}^* \|^2\end{aligned}$$

$$\text{Therefore,} \quad -\sigma \tilde{\mathbf{W}}^T \hat{\mathbf{W}} \leq \frac{\sigma}{2} \| \mathbf{W}^* \|^2 - \frac{\sigma}{2} \left\| \tilde{\mathbf{W}} \right\|^2 \tag{14}$$

$$\left(\frac{r}{\sqrt{\varepsilon} g(\zeta)} - \frac{\sqrt{\varepsilon} d(t)}{2} \right)^2 \geq 0; \Rightarrow \frac{r^2}{\varepsilon g^2(\zeta)} + \frac{\varepsilon d^2(t)}{4} - \frac{r d(t)}{g(\zeta)} \geq 0$$

$$\text{Therefore,} \quad \frac{r d(t)}{g(\zeta)} \leq \frac{r^2}{\varepsilon g^2(\zeta)} + \frac{\varepsilon d^2(t)}{4} \tag{15}$$

$$\left(\frac{r}{\sqrt{2\varepsilon g(\zeta)}} + \frac{\mu \sqrt{\varepsilon g(\zeta)}}{\sqrt{2}} \right)^2 \geq 0; \Rightarrow \frac{r^2}{2\varepsilon g(\zeta)} + \frac{\mu^2 \varepsilon g(\zeta)}{2} + \mu r \geq 0$$

$$\text{Therefore,} \quad -\mu r \leq \frac{r^2}{2\varepsilon g(\zeta)} + \frac{\mu^2 \varepsilon g(\zeta)}{2} \leq \frac{r^2}{2\varepsilon g(\zeta)} + \frac{\mu^2 \varepsilon \bar{g}}{2} \tag{16}$$

As $|\mu| \leq \mu_0$, $|d(t)| \leq d_{0,}$ using (14), (15), and (16), we have:

$$\dot{L} \leq -\frac{r^2}{2\varepsilon g(\zeta)} - \frac{\sigma}{2} \left\| \tilde{\mathbf{W}} \right\|^2 + \frac{\varepsilon \mu_0^2 \bar{g}}{2} + \frac{\varepsilon d_0^2}{4} + \frac{\sigma}{2} \| \mathbf{W}^* \| \tag{17}$$

Considering $\tilde{\mathbf{W}}^T \mathbf{\Gamma}^{-1} \tilde{\mathbf{W}} \leq \bar{\eta} \left\| \tilde{\mathbf{W}} \right\|^2$, where $\bar{\eta}$ is the largest eigenvalue of $\mathbf{\Gamma}^{-1}$, we have:

$$\dot{L} \leq -\frac{1}{\alpha_0} V + \frac{\varepsilon \mu_0^2 \bar{g}}{2} + \frac{\varepsilon d_0^2}{4} + \frac{\sigma}{2} \| \mathbf{W}^* \|^2 \tag{18}$$

where $\alpha_0 = \max \{\varepsilon, \bar{\eta}/\sigma\}$.

Using Lemma B5 in [24],

$$\begin{aligned} L(t) &\leq e^{-t/\alpha_0} V(0) \\ &\quad + \left(\frac{\varepsilon}{2} \mu_0^2 \bar{g} + \frac{\varepsilon d_0^2}{4} + \frac{\sigma}{2} \left\| \mathbf{W}^* \right\|^2 \right) \int_0^t e^{-(t-\tau)/\alpha_0} d\tau \\ &\leq e^{-t/\alpha_0} V(0) \\ &\quad + \alpha_0 \left(\frac{\varepsilon}{2} \mu_0^2 \bar{g} + \frac{\varepsilon d_0^2}{4} + \frac{\sigma}{2} \left\| \mathbf{W}^* \right\|^2 \right), \forall t \geq 0 \end{aligned} \tag{19}$$

Since $L(0)$ is bounded, so from (19), r and $\hat{\mathbf{W}}(t)$ are bounded. From (13), it follows that $L \geq \frac{1}{2} \frac{r^2}{g(\zeta)}$.

Thus, $r \leq \sqrt{2g(\zeta)L} \leq \sqrt{2g(\zeta)L} \leq \sqrt{2\bar{g}L}$.

Using $\sqrt{m+n} \leq \sqrt{m} + \sqrt{n}$ $(m > 0, n > 0)$ in (19), we obtain the bounds on tracking error in (20).

$$\begin{aligned} |r| &\leq e^{-t/2\alpha_0} \sqrt{2\bar{g} L(0)} \\ &\quad + \sqrt{\alpha_0 \bar{g}} \left(\varepsilon \mu_0^2 \bar{g} + \frac{\varepsilon d_0^2}{2} + \sigma \left\| \mathbf{W}^* \right\|^2 \right)^{1/2}, \ \forall t \geq 0. \end{aligned} \tag{20}$$

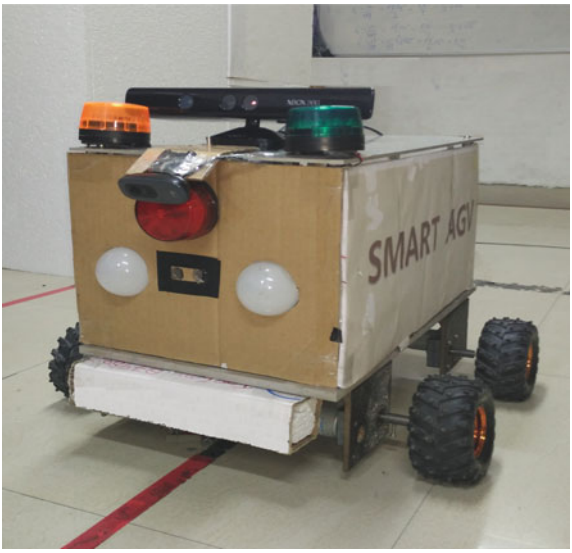


Fig. 5 Automated guided vehicle (at IIT Kanpur) used for algorithm implementation

3 Results and Discussions

The self-developed automated vehicle shown in Fig. 5 has been used for experimental purpose. It is a four-wheeled deferentially driven vehicle with 200 RPM DC geared motors (47.19 kg-cm torque at stall current of 10.6 A). The algorithms are

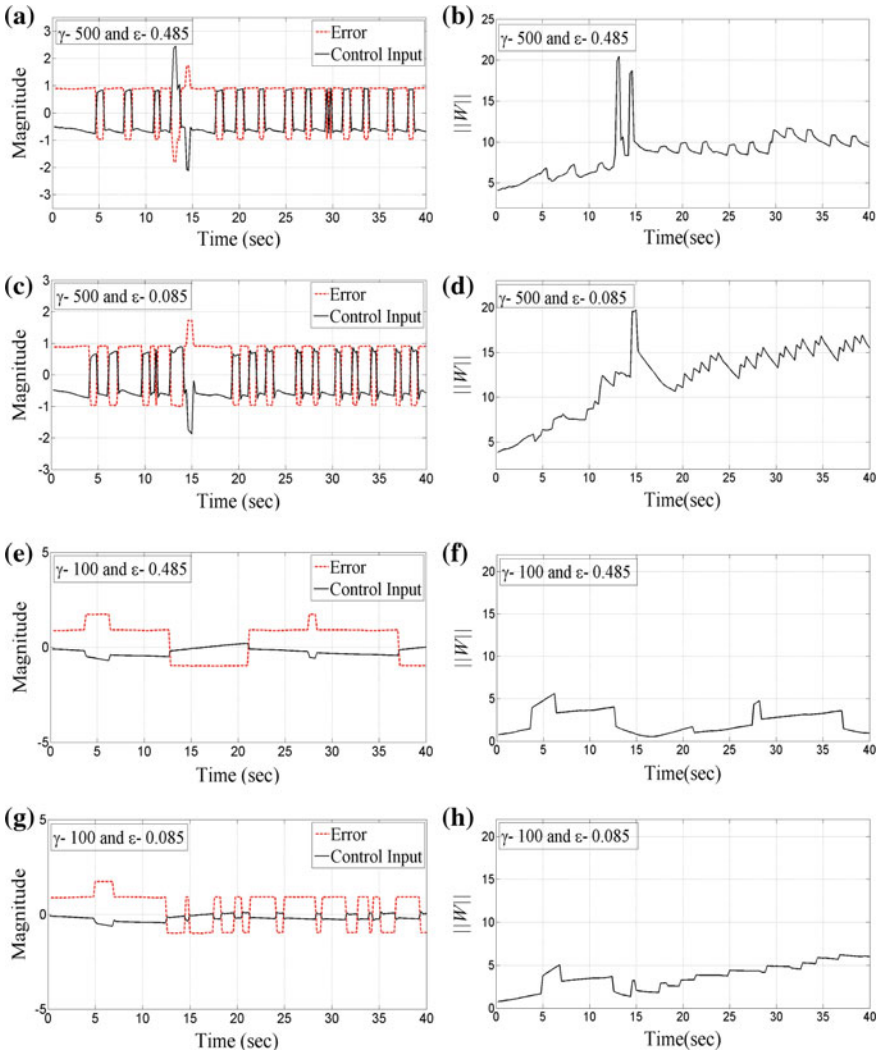


Fig. 6 Control input-error relation and weight in designed controller for different values of γ and ϵ

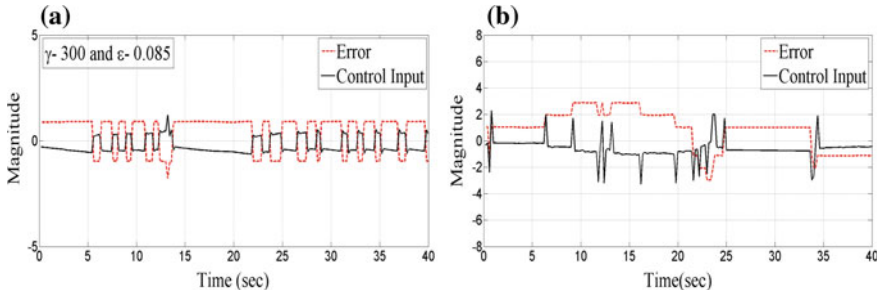


Fig. 7 Error and control input of different controllers: **a** designed controller and **b** controller as [25]

Table 1 Performance with designed controller and [25]

Controllers			Deviation in control input	Mean square control input
Direct adaptive	$\gamma = 500$	$\epsilon = 0.485$	0.6986	0.5647
		$\epsilon = 0.085$	0.6342	0.4684
	$\gamma = 300$	$\epsilon = 0.485$	0.4052	0.2142
		$\epsilon = 0.085$	0.3945	0.2055
	$\gamma = 100$	$\epsilon = 0.485$	0.207	0.0846
		$\epsilon = 0.085$	0.1684	0.061
Controller as [25]			0.7644	0.8159

implemented at remotely placed controller unit with 2.54 GHz processing platform. The automated vehicle has an onboard processing unit of 1.24 GHz for data transfer and reception.

The design parameters used are: NN with network structure 5-20-1(input–hidden–output) nodes and input vector $z = [\zeta_1, \zeta_2, r, \frac{r}{\epsilon}, v]$ is used. The initial weights for the above network are assumed to be zero. λ and σ are taken as 5 and 0.005, respectively. Initial states of RPD and its rate of change are taken as zero, $[\zeta_1, \zeta_2] = [0, 0]$. The desired RPD for our experimental analysis is taken as zero, $\zeta_d = 0$.

The real-time path tracking operation is done for different values of γ and ϵ. The tracking error, control input, and weights for different values are shown in Fig. 7. Figures show bounded nature of tracking error and weights. Table 1 shows standard deviation and mean square values of control effort for the controllers. The direct adaptive controller performs better than indirect adaptive controller [25] as indirect adaptive controller required approximation of plant dynamics for design of control input. Figure 6 and Table 1 also show that control effort and chattering in it can be reduced with the decrease in values of γ and ϵ (also discussed in Remark 1).

4 Conclusions

In this paper, we have designed an NN-based direct adaptive controller with input as RPD from fuzzy system. The proposed approach is used for tracking by automated vehicle in a real-time scenario. The fuzzy system is used to map RPD around the path appropriately and minimizes the vision-based uncertainties in data. The stability of the closed-loop system is guaranteed with appropriate design parameters through Lyapunov stability analysis as well as with the real-time experiment on the vehicle. The effectiveness of proposed control strategy is illustrated through bounded nature of tracking error and mean square control input. Different control techniques can be applied for vision sensor-based plant by following the discussed methodology. The controller can be further improved by inclusion of saturation values of control input beforehand. The proposed methodology can be used for other vision-based applications too.

References

1. Lu, D.H.: Design and Implementation of Computer Controlled AGV. Shanghai Jiao Tong University, Shanghai (2006)
2. Ye, J.: Design and Research of Magnetic Automatic Guided Vehicle Control System. Wuhan University of Technology, Wuhan (2006)
3. Arora, S., Raina, A.K., Mittal, A.K.: Collision avoidance among AGVs at junctions. In: Intelligent Vehicles Symposium Proceedings of the IEEE, pp. 585–589 (2000)
4. Chen, J., Dawson, D.M., Dixon, W.E., Behal, A.: Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension. IEEE Trans. Control Syst. Technol. **13**(5), 814–825 (2005)
5. Fang, Y., Dixon, W.E., Dawson, D.M., Chawda, P.: Homography based visual servo regulation of mobile robots. IEEE Trans. Syst. Man Cybern. **35**(5), 1041–1050 (2005)
6. Mariottini, G.L., Oriolo, G., Prattichizzo, D.: Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. IEEE Trans. Robot. **23**(1), 87–100 (2007)
7. Lopez-Nicolas, G., Guerrero, J.J., Sagues, C.: Visual control of vehicles using two-view geometry. Mechatronics **20**(2), 315–325 (2010)
8. Tsakiris, D., Rives, P., Samson, C.: Extending Visual Servoing Techniques to Nonholonomic Mobile Robots. Lecture Notes in Control and Information Sciences. The Confluence of Vision and Control, pp. 106–117. Springer, Berlin (1998)
9. Miyazaki, F., Masutani, Y.: Robustness of sensory feedback control based on imperfect Jacobian. In: Proceedings of the 5th International Symposium on Robotics Research, pp. 201–208 (1990)
10. Gaspar, J., Winters, N., Santos-Victor, J.: Vision-based navigation and environmental representations with an omnidirectional camera. IEEE Trans. Robot. Autom. **16**(6), 890–898 (2000)
11. Menegatti, E., Maeda, T., Ishiguro, H.: Image-based memory for robot navigation using properties of the omnidirectional images. Robot. Auton. Syst. **47**(4), 251–267 (2004)
12. DeSouza, G.N., Kak, A.C.: Vision for mobile robot navigation: a survey. IEEE Trans. Pattern Anal. **24**(2), 237–267 (2002)
13. Bonin Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: a survey. J. Intell. Robot. Syst. **53**(3), 263–296 (2008)
14. Rajurkar, S.D., Kar, A.K., Goswami, S., Verma, N.K.: Optimal path estimation and tracking for an automated vehicle using GA optimized fuzzy controller. In: 2016 11th International Conference on Industrial and Information Systems (ICIIS), Roorkee, pp. 365–370 (2016)

15. Kar, A.K., Dhar, N.K., Chandola, R., Nawaz, S.S.F, Verma, N.K.: Trajectory tracking by automated guided vehicle using GA optimized sliding mode control. In: 2016 11th International Conference on Industrial and Information Systems (ICIIS), Roorkee, pp. 71–76 (2016)
16. Kar, A.K., Dhar, N.K., Nawaz, S.S.F, Chandola, R., Verma, N.K.: Automated guided vehicle navigation with obstacle avoidance in normal and guided environments. In: 2016 11th International Conference on Industrial and Information Systems (ICIIS), Roorkee, pp. 77–82 (2016)
17. Verma, N.K., Sahu, S.K., Mustafa, A., Ocean, Y., Dhar, N.K., Salour, A.: Priority based optimal path routing for automated guided vehicle. In: IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI), Kanpur, pp. 1–7 (2015)
18. Verma, N.K., Nama, P., Kumar, G., Siddhant, A., Ocean, Y., Raj, A., Dhar, N.K., Salour, A.: Vision based object follower automated guided vehicle using compressive tracking and stereo-vision. In: IEEE Bombay Section Symposium (IBSS), pp. 1–6 (2015)
19. Verma, N.K., Kaushal, A.: Implementation of vehicle to grid concept using ANN and ANFIS controller. In: 9th IEEE Conference on Industrial Electronics and Applications, pp. 960–965 (2014)
20. Verma, N.K., Kaushal, A.: Vehicle to grid concept using ANN and ANFIS Controller. In: IEEE Conference on Industrial Electronics and Applications, China, 9–11 Jun, pp. 960–965 (2014)
21. Ge, S.S., Hang, C.C., Zhang, T.: A direct method for robust adaptive nonlinear control with guaranteed transient performance. Syst. Control Lett. **37**(5), 275–284 (1999)
22. Dhar, N.K., Verma, N.K., Behera, L.: On an integrated approach to networked climate control of a smart home. IEEE Syst. J. **99**, 1–12 (2016)
23. Arslan, A., Kaya, M.: Determination of fuzzy logic membership functions using genetic algorithms. Fuzzy Sets Syst. **118**(2), 297–306 (2001)
24. Krstic, M., Kanellakopoulos, I., Kokotovic, P.V.: Nonlinear and Adaptive Control Design. Wiley-Interscience, NewYork (1995)
25. Kwan, C., Lewis, F.L.: Robust backstepping control of nonlinear systems using neural networks. IEEE Trans. Syst. Man, Cybern. **30**(6), 753–766 (2000)

A Modified Extreme Learning ANFIS for Higher Dimensional Regression Problems

C. P. Pramod, Mahendra Singh Tomar and G. N. Pillai

Abstract Extreme learning adaptive neuro-fuzzy inference system (ELANFIS) is a new learning machine which integrates reduction of computational complexity of extreme learning machine (ELM) concept to ANFIS. ELANFIS uses Takagi–Sugeno–Kang (TSK) fuzzy inference system like ANFIS which gives accurate models. Grid partitioning is used in both ANFIS and ELANFIS which has the disadvantage of curse of dimensionality. In this paper, a modified ELANFIS using sub-clustering for input space partitioning is proposed for higher dimensional regression problems. In the proposed structure, sub-clustering is used for input space partitioning of the network. The cluster centers are used to obtain the premise parameters of the ELANFIS, such that it satisfies the constraints for obtaining distinguishable membership functions. Performance of the modified ELANFIS is compared with ANFIS and ELANFIS for real-world higher dimensional regression problems. The modified ELANFIS overcomes the curse of dimensionality with better interpretability compared to ANFIS and ELANFIS.

Keywords ELANFIS · Sub-clustering · Higher dimensional regression

1 Introduction

Adaptive neuro-fuzzy inference system, ANFIS [1, 2], has universal approximation capability and thus is used extensively for function approximation. Learning capability of neural networks and knowledge representation of fuzzy are combined in ANFIS network. The training of ANFIS is done using a hybrid learning algorithm which is

C. P. Pramod (✉) · M. S. Tomar · G. N. Pillai
Department of Electrical Engineering, Indian Institute of Technology Roorkee, Roorkee, India
e-mail: cppramod.cpp@gmail.com

M. S. Tomar
e-mail: mahendrasinghtomar@gmail.com

G. N. Pillai
e-mail: gopinath.pillai@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_22

computationally expensive. The ELM concept has been applied to neuro-fuzzy systems to overcome these drawbacks [3–5]. Extreme learning ANFIS (ELANFIS) has shown good results for regression problems [5]. The randomness inherent in ELM networks is reduced by ELANFIS network by incorporating the explicit knowledge representation using fuzzy membership functions. In [6], ELANFIS performance for prediction of ground motion parameters associated with seismic signals is compared with conventional methods and is found to perform better than conventional methods. In [7], a zero-order ELANFIS is used for classification problems. In regularized ELANFIS (RELANFIS), the regularization strategy of kernel-based ELM is incorporated in ELANFIS to improve the generalization performance [8]. RELANFIS is compared with SVM, kernel-based regression techniques, and ELM-based networks for regression problems and is found to have similar or better generalization capability. In PSO-RELANFIS, regularization parameter is optimized using PSO technique [9] and has similar or better generalization capability for regression and classification problems. CEEMDAN–ELANFIS incorporates the empirical mode decomposition technique into ELANFIS [10]. CEEMDAN–ELANFIS is used for landslide displacement forecasting [10] and has better accuracy compared to EMD-SVM and ELANFIS. Grid partitioning is used in ANFIS, ELANFIS, and RELANFIS for input space partitioning which suffers from the curse of dimensionality, resulting in extremely high number of rules with increase in number of input features. Thus, these networks cannot be used for higher dimensional problems. As these networks have good function approximation capability, they are widely used for nonlinear system modeling. NARMAX model structure takes past input and output values to predict future output values [11]. For a multiple input multiple output (MIMO) nonlinear system, NARMAX structure will result into a higher dimensional problem and thus ANFIS, ELANFIS, and RELANFIS will be insufficient to obtain accurate models. The clustering of input space leads to lesser number of rules and thus can avoid the effect of curse of dimensionality. There are many clustering algorithms in the literature like fuzzy c-means, progressive clustering, compatible cluster merging algorithms, and other optimization-based clustering techniques [12], but Chiu's subtractive clustering [13] is computationally less expensive as it does not require optimization.

In this paper, a modification in ELANFIS network topology is proposed for avoiding the curse of dimensionality. In this network, the input space partitioning of the TS model fuzzy inference system is done using sub-clustering of input–output training data. The premise parameters are selected randomly around the cluster centers obtained from sub-clustering such that the membership functions are distinct which improves the interpretability of the FIS. Interpretability of a FIS means the ability to represent the system behavior in an understandable manner and depends on various factors like model structure, number of input features, number of rules, distinguishability of membership functions [14, 15]. The consequent parameters are obtained by pseudo-inverse like in ELMs and ELANFIS. With the help of higher dimensional regression problem examples, the paper shows that, compared to ANFIS and ELANFIS, the proposed network reduces the number of rules and thus avoids the curse of dimensionality.

Rest of the paper is organized as follows. Section 2 gives the proposed network topology. In Sect. 3, algorithm for the modified ELANFIS is given. In Sect. 4, the performance of the proposed method for higher dimensional regression problems is compared with that of ANFIS and ELANFIS. Section 5 concludes the performance comparison of ANFIS, ELANFIS, and the proposed network.

2 Proposed Network Topology

The architecture of the network is similar to ELANFIS, the difference being in input space partitioning. ELANFIS and ANFIS use grid partitioning for input space partitioning, whereas the proposed network uses sub-clustering of input–output training data based on Chiu's algorithm [13]. The Takagi–Sugeno (TS) fuzzy inference system is used, and the number of membership functions and number of rules are equal to the number of clusters obtained by the Chui's algorithm. Fuzzy rules will be in the form:

$$\begin{aligned}&\text{Rule } \mathrm{R_i} : \text{if}(x_1 \text{ is } Mf_1^{ci}) \text{ and } (x_2 \text{ is } Mf_2^{ci}) \text{ and } \ldots \text{ and } (x_n \text{ is } Mf_n^{ci}) \text{ THEN } (\text{y is } \beta_i)\\ &\text{where i} = 1, 2, \ldots, \text{ L and L is number of clusters}\end{aligned} \tag{1}$$

$x = [x_1, x_2, \ldots, x_n]^T$ is the crisp n-dimensional input variable. Mf_j^{ci} ($j = 1, 2 \ldots, n$) represent the membership functions corresponding to input variable x_j in the cluster i. β_i is the crisp output for rule i, which is obtained as a linear combination of input variables given by:

$$\beta_i = p_{i0} + p_{i1}x_1 + p_{i2}x_2 + \cdots + p_{in}x_n \tag{2}$$

where p_{ij} ($j = 0, 1, 2, \ldots, n$) are the consequent parameters.

The degree of membership of the input variable x_j for the membership function Mf_j^{ci} in the rule i is given by μ_{jci}. For all rules, the firing strength can be calculated by:

$$w_i(x) = \mu_{1ci}(x_1) \otimes \mu_{2ci}(x_2) \otimes \cdots \otimes \mu_{nci}(x_n) \tag{3}$$

where $\otimes$ indicates 'and' operator of the fuzzy logic. For all the rules, the normalized firing strength can be obtained as:

$$\bar{w}_i(x) = \frac{w_i(x)}{\sum_{k=1}^{L} w_k(x)} \tag{4}$$

The consequent part of the fuzzy rule is obtained by a neural network with p_{ij} as the weight parameters. Thus, the overall output of the network is obtained by:

$$y = \sum_{i=1}^{L} \beta_i \bar{w}_i(x) \tag{5}$$

If the training data has N instances with n attributes in inputs $[X_1\ X_2\ \ldots\ X_n]_{N\times n}$ and one output $\mathbf{T}_{\mathrm{N}\times 1}$, then the output y in Eq. (5) can be equated to the desired target output values for the N samples and can be written as:

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} \bar{w}_1^1 & \bar{w}_1^1 x_1^1 & \cdots & \bar{w}_L^1 x_n^1 \\ \bar{w}_1^2 & \bar{w}_1^2 x_1^2 & \cdots & \bar{w}_L^2 x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \bar{w}_1^N & \bar{w}_1^N x_1^N & \cdots & \bar{w}_L^N x_n^N \end{bmatrix}_{N\times(L*(n+1))} \begin{bmatrix} p_{10} \\ p_{11} \\ \vdots \\ p_{Ln} \end{bmatrix}_{(L*(n+1))\times 1} \tag{6}$$

$$\text{or } \mathbf{T}_{\mathrm{N}\times 1} = \mathbf{H}_{\mathrm{N}\times(\mathrm{L}*(\mathrm{n}+1))} * \mathbf{P}_{\mathrm{L}*(\mathrm{n}+1)\times 1}$$

Here, $\bar{w_1}^i$ is the normalized firing strength of first rule for input data sample i, and x_1^i is the value of first input variable in the ith input data sample. The consequent parameters can be found by computing the matrix $\mathbf{P}$ by:

$$\mathbf{P} = [\,[\mathbf{H}^T\mathbf{H}]^{-1}\mathbf{H}^T]\mathbf{T} \tag{7}$$

2.1 Modified ELANFIS Architecture

Figure 1 shows architecture of modified ELANFIS network. There are five layers: input layer, firing strength computational layer, firing strength normalization layer, consequent parameter layer, and output layer.

Input layer: Nodes of this layer represent an input variable.

Fuzzification layer: This layer consists of L clusters obtained by Chiu's algorithm which represents the L fuzzy rules. Each cluster has n nodes representing the fuzzy membership functions for n inputs. Two-sided Gaussian functions, shown in Fig. 2, are used as the membership functions for this network. The mathematical expression for two-sided Gaussian functions is

$$g\left(x_j, c_{ijL}, \sigma_{ijL}, c_{ijR}, \sigma_{ijR}\right) = \begin{cases} e^{\frac{-\left(x_j - c_{ijL}\right)^2}{2\sigma_{ijL}^2}}, & x_j < c_L \\ 1, & c_L \le x_j \le c_R \\ e^{\frac{-\left(x_j - c_{ijR}\right)^2}{2\sigma_{ijR}^2}}, & x_j > c_R \end{cases} \tag{8}$$

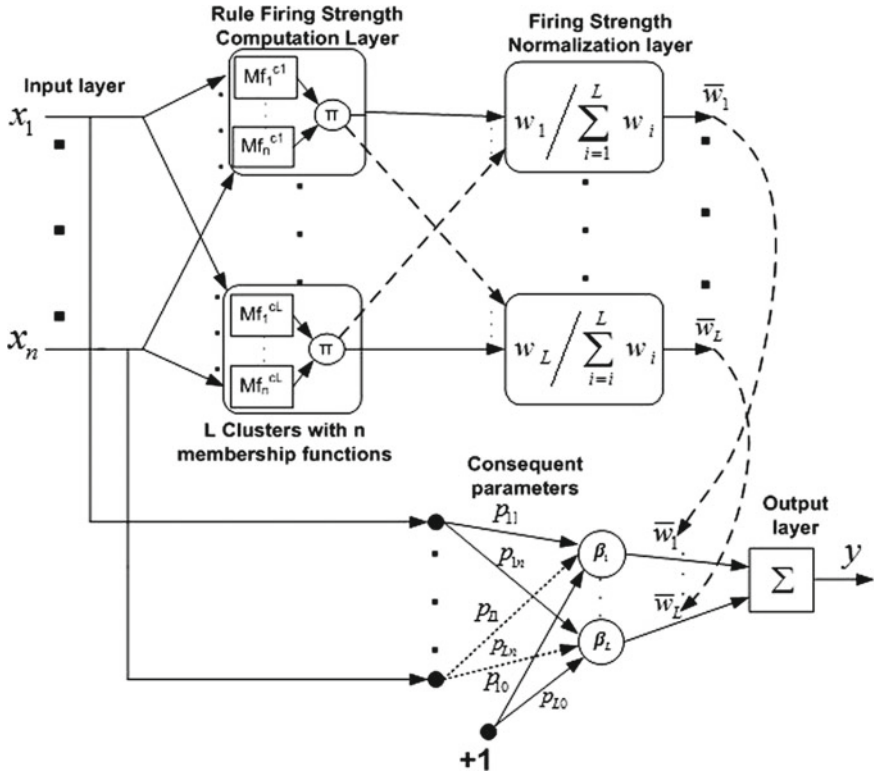


Fig. 1 Modified ELANFIS architecture

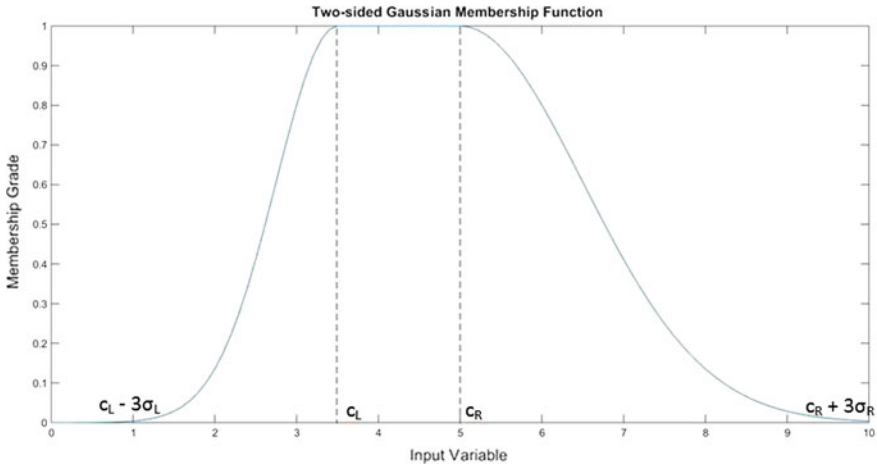


Fig. 2 Two-sided Gaussian membership function

where c_{ijL}, σ_{ijL}, c_{ijR}, and σ_{ijR} are the premise parameters. c_{ijL} and c_{ijR} are the left and right centers, respectively, and σ_{ijL} and σ_{ijR} are the standard deviation on the left side and right side, respectively, of the membership function for the *j*th input variable in *i*th rule. The premise parameters are randomly selected around the cluster centers obtained from Chiu's algorithm. The membership grade μ_{jci}, for the input x_j, is obtained using (8). The firing strength of the rules is obtained by Eq. (3).

Firing strength normalization layer: The firing strength obtained in the previous layer is normalized using Eq. (4).

Consequent parameter layer: This layer consists of a neural network with consequent parameters of fuzzy rules as its weight. The consequent parameters are obtained using extreme learning algorithm. The output of the layer is computed by Eq. (2).

Output layer: The defuzzified output of the overall network is obtained by Eq. (5).

The number of fuzzy rules and membership function depends on the number of clusters obtained by the sub-clustering of input–output data. In the next subsection, Chiu's sub-clustering algorithm is explained.

2.2 Chiu's Sub-clustering Algorithm

Chiu's subtractive clustering algorithm is modification of mountain clustering method which is a class of potential function methods [13, 16]. In mountain clustering method, a grid of data space is made and for all grid points the potential value based on the distances from actual data points is computed. The grid point with most number of data points nearby will have highest potential and is chosen as cluster center. But this method suffers from curse of dimensionality. In Chiu's subtractive clustering algorithm, the data points are taken as center candidates instead of grid points and thus overcome the disadvantages of the mountain clustering method.

Let the set of input–output data be represented by Z^N which consists of N samples, $z_1, z_2, \ldots, z_N$ defined in n + m space, where n is the dimension of input space and m is the dimension of output space. The data samples are normalized, and the potential for a data point z_i is computed as:

$$V_i(z_i, Z^N) = \sum_{j=1}^{N} e^{-\alpha \|z_i - z_j\|^2} \tag{9}$$

where $\alpha = 4/(r_a)^2$ and $i = 1, 2, \ldots, N$
where $r_a > 0$ is the radii parameter that determines the neighborhood radius of each data point. Potentials will be higher for points with a dense neighborhood. The data point with highest potential is selected as the first cluster center. Next, the potential of all the remaining data points is reduced according to their distance from the cluster

center. Denoting z_1^* as the first group center and denoting its potential as V_1^*, the new potential of the remaining points is computed as

$$V_i \leftarrow V_i - V_1^* e^{-\beta \|z_i - z_1^*\|^2} \tag{10}$$

where $\beta = 4/(r_b)^2$
where the constant $r_b > 0$ is the neighborhood radius for sensitive reductions in potential. r_b is selected higher than r_a so that closely spaced clusters are avoided. Usually, $r_b = 1.5\ r_a$. Reduction in potential of the data points nearer to the cluster center will be greater than rest of the points. Now, the data point associated with highest potential is taken as the next cluster center. This process of obtaining new cluster center and reducing potential is repeated till the potential, and the distance between clusters falls below threshold values. The number of clusters depends on the radii parameter. A small radius will lead to a high number of clusters, and a higher radius will lead to a smaller number of clusters.

3 Modified ELANFIS Learning Algorithm

The main modifications made in the ELANFIS algorithm are the input space partitioning and the inclusion of constraints on premise parameters for maintaining distinguishability of the membership functions. As mentioned in the earlier section, Chiu's sub-clustering algorithm is used for input partitioning and determination of premise parameters. As in ELANFIS, the consequent parameters are obtained using pseudo-inverse. Considering a training data with N instances with *n* attribute inputs $[X_1\ X_2\ \ldots\ X_n]_{N\times n}$ and output data $\mathbf{T}_{N\times 1}$, the steps involved in the learning process are:
Step 1. The Chiu's sub-clustering algorithm is used on the input–output data Z^N to obtain the L clusters and cluster centers, $z_1^*, z_2^*, \ldots, z_L^*$ in n+m dimension, where n denotes the number of inputs and m the number of outputs. Thus, extracting the first n dimension of the cluster center, we obtain the cluster centers for input space, x_1^*, $x_2^*, \ldots, x_L^*$ where $x_i^* = (x_{1i}^*, x_{2i}^*, \ldots, x_{ni}^*)$. The standard deviation is obtained as σ_{ji}^* for *j*th input variable in cluster i.
Step 2. Initially, the Gaussian functions are taken as membership functions and thus the premise parameters are the center and the standard deviation of the Gaussian function. The initialization of the premise parameters is done as:

$$c_{ij} = x_{ji}^* + (-1 + 2 * r_1) * 0.1 * (range_input_j) \tag{11}$$

$$\sigma_{ij} = (0.8 + 0.4 * r_2) * \sigma_{ji}^* \tag{12}$$

where c_{ij} and σ_{ij} are the parameters of the membership function for *j*th input in the *i*th cluster, i = 1, 2, …, L and j = 1, 2, …, n. r_1 and r_2 are random values in between 0 and 1.

Thus, membership functions parameters for all input variables are initialized.

Step 3. All membership function of the first input variable is considered, and the centers are sorted in increasing order. Starting from smallest center, the distance between the adjacent centers is calculated and if the difference is less than threshold value, the larger center of two is moved away from the smaller one. The new position of the center is found by:

$$c_{(k+1)j}^{new} = c_{kj} + c_{th} * (range_input_j) \tag{13}$$

where c_{kj} is kth center after sorting for input j and $c_{(k+1)j}^{new}$ is the new center for the right neighbor of kth center. cth is the threshold value obtained as:

$c_{th} = 0.1$
while (1/no of rules) $< c_{th}$

$c_{th} = c_{th}/2$

end while

This is repeated for membership functions of all input variable.

Step 4. In this step, the premise parameters are tuned to satisfy the constraints to obtain distinguishable membership functions. First, all membership functions are converted from the Gaussian function to two-sided Gaussian functions. As mentioned in the earlier section, the parameters of two-sided Gaussian function are given by $[c_{ijL}, \sigma_{ijL}, c_{ijR}, \sigma_{ijR}]$ for input variable j in cluster i. For maintaining interpretability of FIS, overlapping of adjacent membership functions can be restricted to a limit as it becomes difficult to distinguish between the membership functions in case its 'right zero' goes beyond the right zero of the function in the right side and similarly for the left side of the functions [16]. The right zero occurs at $(c_{ijR} + 3 * \sigma_{ijR})$, and the left zero occurs at $(c_{ijL} - 3 * \sigma_{ijL})$. Thus, the constraints for maintaining interpretability can be defined as:

$$\begin{aligned} c_{kR} + 3 * \sigma_{kR} &\leq c_{iR} + 3 * \sigma_{iR} \\ c_{kL} - 3 * \sigma_{kL} &\leq c_{iL} - 3 * \sigma_{iL} \\ c_{iL} - 3 * \sigma_{iL} &\geq c_{jL} - 3 * \sigma_{jL} \\ c_{iR} + 3 * \sigma_{iR} &\geq c_{jR} + 3 * \sigma_{jR} \end{aligned} \tag{14}$$

where i and j represent the right and left neighbor functions, respectively, of any membership function k. Starting from the membership functions in the leftmost end, all membership functions for all the input variables are checked for whether they satisfy the constraints for overlapping or not. If it does not satisfy the constraints, the standard deviation of function is changed as:

$$\sigma_{kL}^{new} = \frac{c_{jL} - 3 * \sigma_{jL} - c_{kL}}{-3} \tag{15}$$

Step 5. With the premise parameters obtained in the previous step, the membership grade for input variables is calculated using Eq. (8). The firing strength of rules is computed using Eq. (3).
Step 6. The normalized firing strength of rules is computed using Eq. (4). These act as the weights of the neural network in the output layer of modified ELANFIS.
Step 7. In the output Eq. (5), the output y is equated to the desired target output values and the matrix Eq. (6) is obtained. The consequent parameters are found using Eq. (7).

The steps are repeated for 50 trials.

4 Simulation Examples

For performance comparison of the modified ELANFIS with ANFIS and ELANFIS for higher dimensional regression problems, a few real-world regression dataset examples have been simulated using MATLAB. Comparison of NARMAX model for a MIMO system is also done where data for modeling is generated by simulating the mathematical model of the system. In the following subsections, the simulation results are given.

4.1 Real-World Regression Problems

For performance comparison of ANFIS, ELANFIS, and the proposed network, simulations are done in MATLAB 2014b on an Intel Core i7 Processor with 8 GB RAM for four real-world regression problems, servo dataset, stock domain dataset, Istanbul stock exchange dataset, and kinematics of robotic arm dataset [17].

The servo dataset consists of 4 input features and one output and has 167 instances. Istanbul stock exchange dataset is a regression dataset which has 7 input features and 536 instances. Stock domain dataset has 950 instances for 9 input features. Kinematics of robotic arm dataset consists of 8 input features and 8192 instances. Two-third of the data is used for training, and one-third is used for testing. Figure 3 shows the training and testing root-mean-square error for the simulations. The simulation results are presented in Table 1.

From Table 1, it can be seen that for servo dataset with increase in number of rules, the training time of ANFIS increases exponentially. Due to integration of ELM concept into the network, the training time of ELANFIS does not increase as much as in case of ANFIS even though there is huge increase in number of rules with increase in membership functions. Modified ELANFIS performs better than ANFIS and ELANFIS with lesser number of rules and faster training time. For Istanbul stock exchange dataset, ANFIS training results in system going out of memory when three or more membership functions are used. The number of rules and training time for ELANFIS rises significantly with increase in number of membership

Table 1 Performance comparison for real-world regression problems

Dataset	Algorithm	No. of rules	Training time	Training error (RMSE)	Testing error (RMSE)
Servo dataset	ANFIS	16	0.2845	0.1827	1.1873
		81	5.1354	3.9006e−06	0.7860
		625	1028.737	1.7855e−07	1.5357
	ELANFIS	16	0.1238	0.1674	1.5179
		81	0.3003	0.0000	0.7679
		625	2.0612	0.0000	1.7378
	Modified ELANFIS	2	0.0343	0.7826	1.4872
		3	0.0416	0.6216	1.4595
		5	0.0552	0.6200	0.7961
		6	0.0689	0.4768	0.7572
Istanbul stock exchange dataset	ANFIS	128	269.61	0.006376	0.04635
	ELANFIS	128	2.000	0.0000	9.5513
		2187	116.78	0.0000	4.8596
		78,125	39904.74	0.0000	0.7611
	Modified ELANFIS	2	0.0567	0.013	0.0149
		3	0.0816	0.0127	0.0158
		5	0.1225	0.1427	0.1225
Stock domain dataset	ANFIS	512	16106.294	0.0048	1.781734
	ELANFIS	512	52.5645	0.0604	8.9695
		19,683	8542.3431	0.0000	5.4149
	Modified ELANFIS	5	0.29975	0.9213	0.9218
		7	0.3346	0.7948	0.8045
		9	0.38427	0.7526	0.8111
		11	0.4663	0.6820	0.7514
Kinematics of robotic arm dataset	ANFIS	256	23320.22	0.0548	0.1299
	ELANFIS	256	364.4108	0.0604	0.1897
		6561	27041.85	0.0000	1.5361
	Modified ELANFIS	6	2.1109	0.1704	0.1725
		9	2.6648	0.1603	0.1635
		15	3.7973	0.1424	0.1447
		21	5.2123	0.1309	0.1362

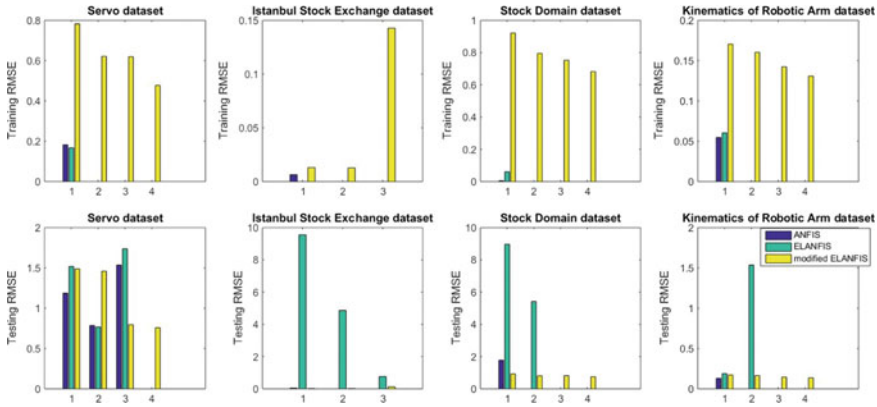


Fig. 3 Performance comparison of ANFIS, ELANFIS, and modified ELANFIS for real-world regression problems

functions. For this dataset as well, modified ELANFIS performs better than other two networks. For stock domain dataset, ANFIS and ELANFIS result in system going out of memory for larger number of rules. In this case as well, the training time is very small for modified ELANFIS compared to the other two networks. Better accuracy is also achieved with lesser number of rules with modified ELANFIS. Kinematics of robotic arm dataset is a large dataset and thus results in very high training time for ANFIS and ELANFIS networks. The system goes out of memory for higher number of membership functions in case of ANFIS and ELANFIS. The training time is significantly smaller for modified ELANFIS compared to the other two networks.

4.2 Modeling of MIMO System

In this subsection, NARMAX models for a MIMO system, wheeled mobile robot, are obtained using ANFIS, ELANFIS, and modified ELANFIS and the accuracy of the models is compared. Wheeled mobile robot is a nonlinear mechanical system with nonholonomic constraints. The dynamic equation of motion for the system [18] is given by:

$$M(q)\ddot{q} + V(q, \dot{q}) = E(q)\tau - A^T\lambda \tag{16}$$

where q is generalized coordinates, τ is the input vector, λ is the vector of constraint forces, M(q) is the inertia matrix, $V(q, \dot{q})$ is the centripetal matrix, E(q) is the input transformation matrix, and A(q) is the matrix associated with the constraints. The generalized coordinates are position coordinates x_c and y_c, the orientation of mobile robot Φ, and angular position for the left and right driving wheels θ_L and θ_R.

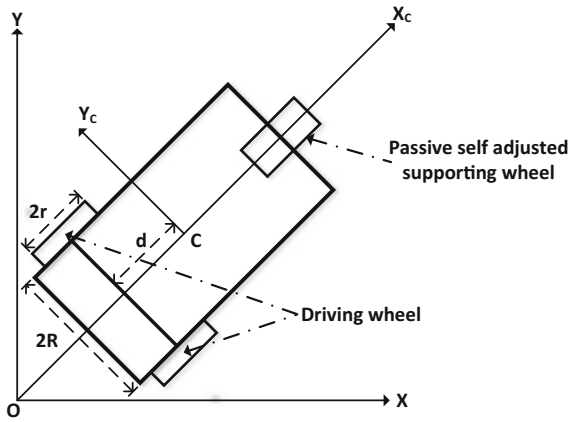


Fig. 4 Two-wheel mobile robot

Figure 4 shows the two-wheel mobile robot. The data for the mobile robot is generated in MATLAB by simulating the dynamic equation given by Eq. 16. In this paper, NARMAX models for the left and right wheel velocities are obtained. The inputs to the models are $[\tau_R(k), \tau_L(k), \tau_R(k-1), \tau_L(k-1), \omega_R(k-1), \omega_L(k-1)]$, where $\tau_R(k)$ and $\tau_L(k)$ are the torque given to right wheel and left wheel, respectively, at kth instant, and $\omega_R(k-1)$ and $\omega_L(k-1)$ are right wheel and left wheel velocities, respectively, at $(k-1)$th instant. $\omega_R(k)$ and $\omega_L(k)$ are taken as the output of the two models. Models are obtained using ANFIS, ELANFIS, and modified ELANFIS. The comparison of performance is given in Table 2. From Table 2, it can be observed that ANFIS and ELANFIS result in system going out of memory for larger number of rules. Modified ELANFIS model has better accuracy and employs less number of rules and faster training compared to other two networks.

Since membership function distinguishability is maintained in the proposed network, the interpretability of the network is better than the other two networks resulting in more understandable model.

5 Conclusion

The modified ELANFIS makes use of the Chiu's sub-clustering algorithm for input space partitioning. The premise parameters of the fuzzy rules are tuned to satisfy the constraints for distinguishability of membership functions. The proposed modifications thus avoid the curse of dimensionality and can be used for higher dimensional problems. Comparison of the modified ELANFIS with ANFIS and ELANFIS for real-world higher dimensional regression problem is performed via simulation in MATLAB. Simulation results show that the modified ELANFIS performs better than ANFIS and ELANFIS. Modified ELANFIS results in smaller number of rules, has faster training, and achieves better accuracy. NARMAX models for a MIMO

Table 2 Performance comparison for NARMAX modeling of mobile robot

NARMAX model	Algorithm	No. of rules	Training time	Training error (RMSE)	Testing error (RMSE)
Right wheel velocity model	ANFIS	64	110.9004	0.03181	0.04607
	ELANFIS	64	4.6975	0.0311	0.0869
		729	364.7643	0.0022	2171.9120
	Modified ELANFIS	3	0.9726	0.04730	0.0511
		4	0.5121	0.0432	0.0463
		7	0.7045	0.0396	0.0432
Left wheel velocity model	ANFIS	64	111.2651	0.03268	0.04582
	ELANFIS	64	4.79981	0.0321	0.5749
		729	366.2605	0.0022	1750.227
	Modified ELANFIS	3	0.3776	0.0487	0.0503
		4	0.5136	0.04185	0.04400
		7	0.5700	0.04034	0.04370

system, mobile robot, are obtained using ANFIS, ELANFIS, and modified ELANFIS, and it is shown that modified ELANFIS model performs better than other two network models. Since membership function distinguishability is maintained in modified ELANFIS, the model obtained using modified ELANFIS is more interpretable than other two models. This makes the proposed network suitable for system modeling and control of higher order nonlinear systems. Optimization of the number of clusters obtained using Chiu's algorithm can be done as future work.

References

1. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference systems. IEEE Trans. Syst. Man Cybern. **23**(3), 665–685 (1993)
2. Jang, J.-S.R., Sun, C.-T.: Neuro-fuzzy modeling and control. Proc. IEEE **83**(3), 378–406 (1995)
3. Sun, Z.-L., Au, K.-F., Choi, T.-M.: A neuro-fuzzy inference system through integration of fuzzy logic and extreme learning machines. IEEE Trans. Syst. Man Cybern. B Cybern. **37**(5), 1321–1331 (2007)
4. Rong, H.-J., Huang, G.-B., Sundararajan, N., Saratchandran, P.: Online sequential fuzzy extreme learning machine for function approximation and classification problems. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **39**(4), 1067–1072 (2009)
5. Pillai, G.N., Pushpak, J., Nisha, M.G.: Extreme learning ANFIS for control applications. In: IEEE Symposium on Computational Intelligence in Control and Automation (CICA), pp. 1–8 (2014)
6. Tushar, A., Pillai, G.N.: Extreme learning ANFIS for classification problems. In: 2015 1st International Conference on Next Generation Computing Technologies (NGCT), pp. 784–787. IEEE (2015)

7. Thomas, S., Pillai, G.N., Pal, K., Jagtap, P.: Prediction of ground motion parameters using randomized ANFIS (RANFIS). Appl. Soft Comput. **40**, 624–634 (2016)
8. Shihabudheen, K.V., Pillai, G.N.: Regularized extreme learning adaptive neuro-fuzzy algorithm for regression and classification. Knowl. Based Syst. **127**, 100–113 (2017)
9. Shihabudheen, K.V., Mahesh, M., Pillai, G.N.: Particle swarm optimization based extreme learning neuro-fuzzy system for regression and classification. Expert Syst. Appl. **92**, 474–484 (2018)
10. Shihabudheen, K.V., Pillai, G.N., Bipin Peethambaran, : Prediction of landslide displacement with controlling factors using extreme learning adaptive neuro-fuzzy inference system. Appl. Soft Comput. **61**, 892–904 (2017)
11. Narendra, K.S., Parthasarathy, K.: Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Netw. 4–27 (1990)
12. Dave, R.N., Krishnapuram, R.: Robust clustering methods: a uni/ed view. IEEE Trans. Fuzzy Syst. **5**, 270–293 (1997)
13. Chiu, S.L.: Fuzzy model identification based on cluster estimation. J. Intell. Fuzzy Syst. **2**(3), 267–278 (1994)
14. Bodenhofer, U., Bauer, P.: A formal model of interpretability of linguistic variables. In: Interpretability Issues in Fuzzy Modeling, pp. 524–545. Springer, Berlin, Heidelberg (2003)
15. Casillas, J., et al.: Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview. In: Interpretability Issues in Fuzzy Modeling, pp. 3–22. Springer, Berlin, Heidelberg (2003)
16. Paiva, R.P., Dourado, A.: Interpretability and learning in neuro-fuzzy systems. Fuzzy Sets Syst. **147**(1), 17–38 (2004)
17. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. Dept. Inf. Comput. Sci., Univ. California, Irvine, CA. http://archive.ics.uci.edu/ml/datasets.html (1998)
18. Yamamoto, Y., Yun, X.: Coordinating locomotion and manipulation of a mobile manipulator. In: Proceedings of the 31st IEEE Conference on Decision and Control (1992)

Part V
Object Identification and Scene Analysis

Analysis of Transfer and Residual Learning for Detecting Plant Diseases Using Images of Leaves

Ines Khandelwal and Sundaresan Raman

Abstract The study of plant diseases is critical for alleviating the problem of food security all over the world. The most critical step in mitigating this problem is the correct and appropriate timely identification of the disease. The first step in identification of a disease is visual inspection. The massive scale of this problem and lack of professionals create a need for a automated accurate visual inspection technique. Recent advances in the field of computer vision, primarily through techniques such as use of convolutional neural networks and deep learning have generated impressive results in the field of image classification and object recognition. In this paper, we address the problem of detecting plant diseases using images of leaves using different state-of-the-art approaches. We use the Plant Village dataset comprising of 86,198 images of 25 crops across 57 classes (healthy and specific diseases). The images are of high quality and have been taken manually under appropriate lighting conditions. On this dataset, our model is able to attain a significantly high average accuracy of 99.374% using transfer learning on state-of-the-art models trained on the ILSVRC 2012 dataset having 1.2 million images across 1000 classes.

Keywords Deep learning · Neural networks · Residual learning · Transfer learning · Plant diseases

1 Introduction

Addressing the food security of the ever-growing population on earth is one of the most important challenges facing the world today. Food security all over the globe is threatened due to plant diseases which are difficult to control because the population

I. Khandelwal · S. Raman (✉)
Department of Computer Science and Information Systems,
BITS Pilani, Pilani Campus, Pilani, India
e-mail: sundaresan.raman@pilani.bits-pilani.ac.in

I. Khandelwal
e-mail: ines.khandelwal@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_23

of pathogens are variable in time, space, and genotype, and they may also develop resistances to existing methods and evolve creating the need for speedy and accurate identification of plant diseases [1]. It is also possible for new diseases to occur in places where no such past incidents were previously recorded, and thus no local expertise exists among the farming community to combat them [2]. An automated system designed to help identify plant diseases by the plant's appearance and visual symptoms by leveraging advances in the field of computer vision presents an opportunity to expand and enhance the practice of precise plant protection and extend the market of computer vision applications in the field of precision agriculture [3].

1.1 Transfer Learning

Deep convolution neural networks require millions of labeled images for appropriate training. For a large number of practical applications, including ours, such a huge labeled dataset is not easily available. A solution to this problem is to use transfer learning. Transfer learning involves training on two different datasets: a base dataset comprising of large number of labeled images and a target dataset which is the dataset of interest. As per Yosinski et al. [4], in transfer learning, we first train a base neural network on a base dataset and task, which ensures that the network is properly trained and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset.

One of the most popular choices for base dataset is the ImageNet dataset [5]. In Huh et al. [6], authors have discussed what makes ImageNet good for transfer learning and discussed the well-accepted thoughts in the computer vision community that this superior performance could be attributed to the large number of images (1.2 Million) and/or to large number of labeled classes (1000).

As per standard practices, there are two ways to approach transfer learning:

- **Using the base neural network as a fixed feature extractor**
  In this case, the images of the target dataset are fed to the neural network and the features that generate as input to the final classifier layer are extracted. Through these features a new classifier is built and the model is created.
- **Fine Tuning the base network**
  In this case, the final classifier layer is replaced, just like the above case but using backpropagation algorithms, the weights of previous layers are also modified.

A critical question is whether transfer learning is able to achieve better results than training a model from scratch. Yosinski et al. [4] have examined the reasons behind the better results seen in transfer learning and documented that the transferability of features is dependent upon the similarity between the base task and target task. As this similarity increases, using the weights and biases of trained neural networks will be more likely to outperform training using the target task from scratch. Even in case of limited similarity, training from scratch which uses randomly initialized weights

in the beginning will tend to perform worse than training by using the weights of models trained using the base task.

1.2 Residual Learning

He et al. [7] analyzed the question of whether simply stacking layers in a neural network and thereby increasing their depth is guaranteed to give a better accuracy. They concluded through experiments that as the networks get deeper, accuracy gets saturated and then decrease rapidly. This is not caused due to overfitting but due to difficulties in optimizing the layers even after using batch normalization. In order to solve this problem of degrading accuracy and use deeper networks, the authors proposed use of residual learning. The central idea behind residual learning is to add skip connections that bypass a few convolution layers at a time. Each bypass gives rise to a residual block in which the convolution layers predict a residual that is added to the block's input tensor.

Mathematically, this can be formulated as (consider Fig. 1)
Let the input be x.
Let the operation of the stacked layers be $F(x)$.
Then after the addition step the function as an input the Rectified linear Unit (ReLU) $H(x)$ becomes:
$H(x) = F(x) + x$

As per [7] this combination of input and output of convolution layer is easier to optimize and solves the problem of degrading accuracy. Addition of simply the input also does not increase the complexity of the model as it will have to be computed anyhow, and they empirically observe that their proposed residual learning mechanism is able to adapt well to increasing depth by converging faster and also showing an increase in accuracy in contrast to conventional convolutional neural networks.

Using the techniques of transfer learning and residual learning, we attempt to analyze the problem of detecting plant diseases for producing a model with high accuracy and of practical usage. The rest of the paper is organized as follows: In Sect. 2, we describe the dataset using a few representative images. In Sect. 3, we discuss previous works in the field of detection of plant diseases from images of leaves. Subsequent sections detail our experimental setup, results obtained, and inferences that can be drawn from them.

2 Dataset Description

We have used images from the Plant Village project [8]. As of December 26, 2016, we were able to access 86,198 images of 25 crops across 57 classes, labeled with the name of the crop and whether it is healthy or the name of the disease it is suffering from. This dataset is referred to as the Plant Village dataset in the subsequent sections. The

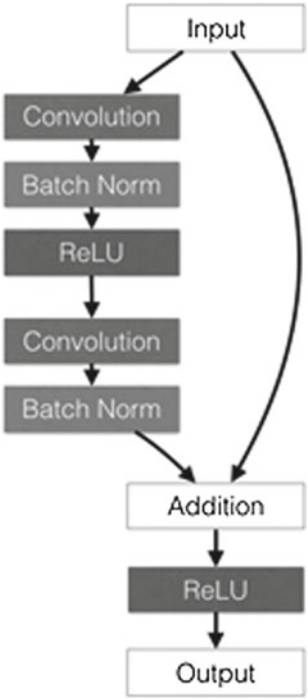


Fig. 1 Residual learning

images in this dataset have been taken by technicians with a standard digital point and shoot camera (Sony DSC—Rx.100/13 20.2 MP) using the automatic mode [8]. The creators of the dataset intentionally tried to ensure that the images were taken under full outside light with a variety of conditions which would resemble the experience of the end user: a farmer using a smartphone [8] (Fig. 2).

The identity of the diseases was confirmed by two experienced plant pathologists who worked with the technicians providing the diagnosis. The diseased crops were obtained either by infecting the crop directly using standard experimental approaches in plant pathology or through natural occurrence in sentinel plots [8].

3 Related Work

Application of deep learning for detecting plant diseases is still in its nascent stages. One of the biggest challenges to developing accurate practical solutions in this area has been the lack of availability of comparatively large datasets as required by the state-of-the-art deep learning techniques. Majority of the work has been specific to a single crop. There are works that have analyzed different deep learning techniques for specific crops such as Banana [9], Tomato [10], Cassava [11].

Fig. 2 Few representative images from dataset

For our specific problem of working with a large dataset comprising of multiple crops, there are fewer works. In Sladojevic et al. [12], authors used 3,000 original images taken from the available Internet sources and extended them to more than 30,000 using appropriate transformation and by using AlexNet achieving a final overall accuracy of the trained model was 96.3%. However, their dataset is not accessible for further experimentation. A related work by Mohanty et al. [13] who worked on a smaller previous version of the Plant Village dataset, [13] had concluded that the best performance was achieved using transfer learning instead of training a neural network from scratch. The difference in test accuracy between using model trained from scratch and the transfer learning model increased as the models was trained on a lesser part of data and tested on the larger part, substantiating that in a real-world example transfer learning is expected to outperform training from scratch. The best-performing model of Mohanty et al. [13] was achieved by transfer learning on GoogleNet [14] by just resetting the weights of the classifier layers, which is same as using the CNN as a fixed feature extractor.

4 Experimental Setup

In our experiments, we use the popular ImageNet dataset as the base dataset and the Plant Village dataset as the target dataset. The images in the ImageNet data are of natural and physical objects and not artificial ones like cartoons or line art, and

the Plant Village dataset has natural images too. Owing to similar nature of these images, the features should be easily transferable [4] which has been verified by the results of Mohanty et al. [13] on a smaller version of the Plant Village dataset, and hence we decided to use the ImageNet dataset as our base dataset.

We use two different strategies for training our models:
In the first one, we follow the strategy of Mohanty et al. [13], who had achieved their best results on a smaller version of Plant Village dataset using GoogleNet [14]. We use the methodology of their best result of using transfer learning by treating the neural network as a fixed feature extractor but use the deeper and newer version of GoogleNet, the Inception V3 Architecture [15] which achieves a better accuracy on the ImageNet dataset than GoogleNet (67.9% accuracy on ImageNet dataset by GoogleNet versus 76.9% by Inception V3). The model trained on base dataset is downloaded from the official Tensorflow repository and the Tensorflow Framework is used for running the experiments.

In the second strategy, we use a combination of residual learning and transfer learning, we use the architecture of ResNet 50 Model proposed in [7] which delivers accuracy of 75.3% on the ImageNet dataset. However, to take full advantage of residual learning and see its implications we do not stop at retraining just the last layer but also fully train the ResNet model through backpropagation. This backpropagation is done by minimizing the negative log-likelihood loss function. The ResNet 50 trained model is obtained from its Torch Implementation and Torch Framework is used to run the experiments.

The models are trained with a batch size of 25. Thus, in every iteration 25 images at random are taken and used to decide the gradient. Images in every class are split randomly into two sets using the chosen training testing percentage. After every epoch (passing the entire dataset to the CNN model), the accuracy of the model is evaluated on the test dataset. The models are studied across two parameters: learning rate (rate at which weights and biases are updated after every iteration) and number of epochs. In order to perform our experiments, we split the Plant Village dataset into two parts at random: training and testing. The percentage of training–testing is varied in the following manner: 20-80, 40-60, 60-40, and 80-20.

5 Results and Discussion

See Table 1.

5.1 Variation in Learning Rate

A neural network can be thought of as finding the appropriate solution in the feature space of its weights and biases and coming closer to those optimal values will increase the accuracy. Failing to attain a high accuracy means that the current set of weights

Table 1 Test accuracy for different experiment configurations with learning rate of 0.01 and batch size of 25

Test percentage	Model name	Epoch number			
		1	5	10	15
Test 20 Train 80	Inception	85.83	89.44	90.87	91.99
	ResNet	97.294	98.65	99.235	99.316
Test 40 Train 60	Inception	83.65	92.1	93.98	94.79
	ResNet	96.577	98.336	99.129	99.167
Test 60 Train 40	Inception	80.45	90.76	93.1	94.11
	ResNet	95.261	98.431	98.876	98.474
Test 80 Train 20	Inception	70.64	87.17	90.32	91.77
	ResNet	93.069	96.859	98.397	98.6479

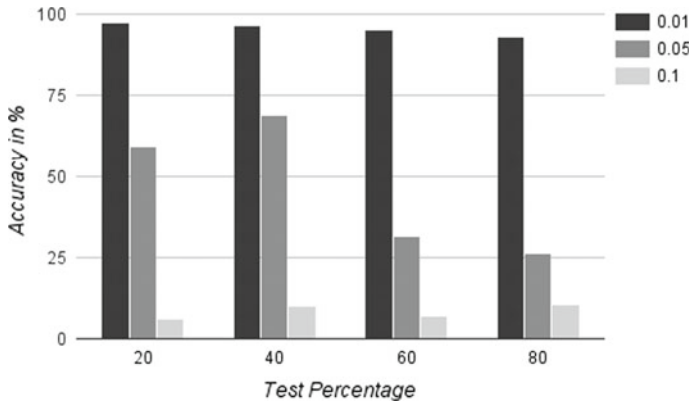


Fig. 3 Substantial changes in accuracy of ResNet 50 in the first epoch as learning rate varies from 0.01 to 0.1

and biases chosen by the model are far away from an optimal solution which the other models are able to come closer to.

Since we use backpropagation and thereby fine-tune the entire network in experimentations on ResNet 50, in the first epoch with a high learning rate the model performs worse as shown in Fig. 3 in contrast to Inception V3 as shown in Fig. 4. However in the subsequent epochs, it is able to attain a higher accuracy as shown in Fig. 5. In terms of finding a solution, the model first is close enough to an optimal value but due to higher learning rate the values of weights and bias shift rapidly from their initial value which was closer to the optimal value and as this model over subsequent epochs make updates to its weight and biases, it starts coming back to better accuracies.

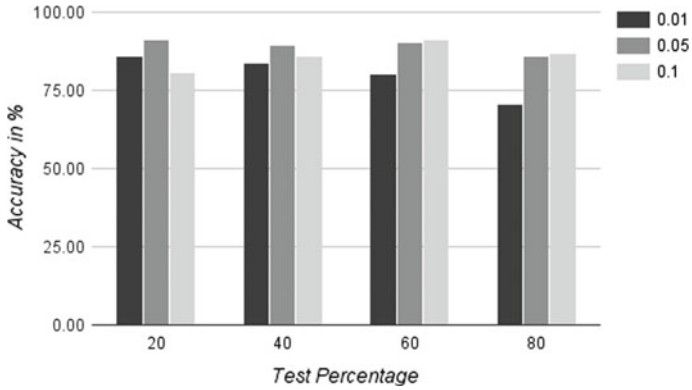


Fig. 4 Comparatively less change in accuracy of inception V3 in the first epoch as learning rate varies from 0.01 to 0.1

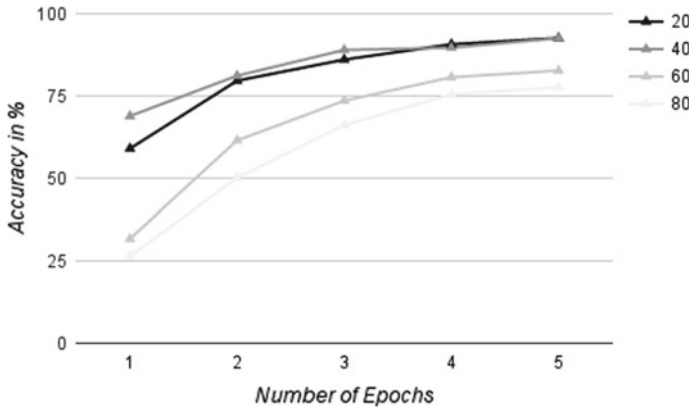


Fig. 5 Accuracy increases as number of epochs increase for ResNet 50 with learning rate of 0.05

5.2 Variation in Number of Epochs

Figures 6 and 7 display a similarity of trend (values in Y-axis change but the shape of graph stays almost the same) in training both ResNet 50 and Inception V3 with learning rate of 0.01 across different test percentages for the first 10 epochs. Though the reduction in training data changes the accuracy of the model, the general shape of the graph stays almost the same implying that in a real-world situation where the test data is large in comparison to the training data these models might have lower accuracy but will display a similar trend. Another key point is that ResNet 50 trains faster as is expected of residual learning models, given its ability to converge faster and thereby saturating faster as well. Thus, increasing epochs has less impact on ResNet 50, whereas the same is not true for Inception V3 which takes a considerably higher time to train fully as well. Moreover, in all these cases, we also find that the

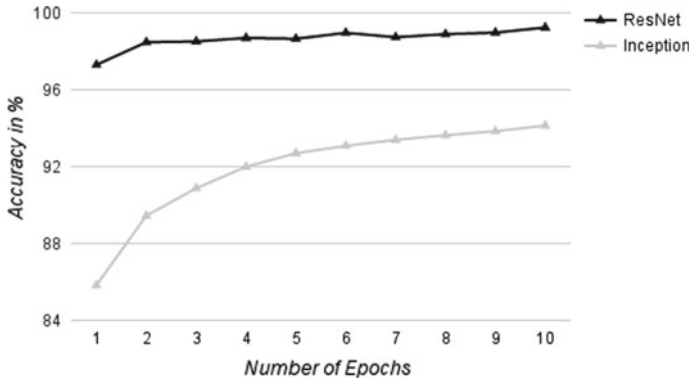


Fig. 6 Comparison of test accuracy with 80-20 train–test split

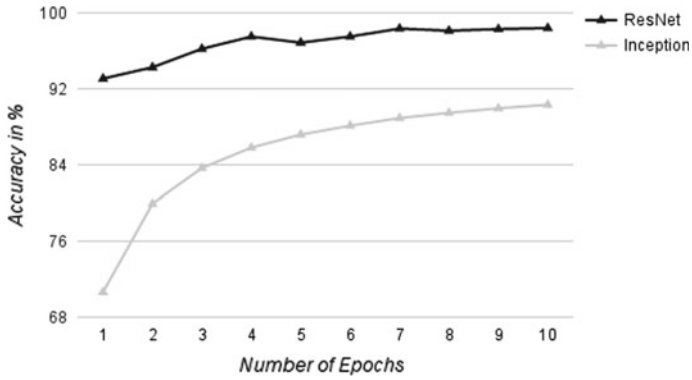


Fig. 7 Comparison of test accuracy with 20-80 train–test split

accuracy attained by ResNet 50 in the first epoch beats the one obtained by Inception V3 after 10 epochs.

5.3 Model Accuracy

Throughout all the rows of Table 1, we are able to clearly see that ResNet 50 is able to outperform Inception V3. This is visible in terms of higher accuracy in the initial few epochs as well as an overall higher accuracy at the end of 15 epochs. Our technique of using a combination of residual learning and transfer learning is able to outperform the kind of strategy proposed in Mohanty et al. [13].

6 Conclusion

6.1 Choice of Model

Automated detection of diseases in plants using images of leaves is still an open problem. In our study, we were able to achieve a very high accuracy of 99.374% using residual learning and transfer learning with a standard 80-20 train–test split. We also see that improvements in accuracy of neural networks when the learning rate is high and the weights and bias are off by considerable margins and can be achieved by increasing the number of epochs. Though the model may end up converging, chances are that the most optimal solution might be missed. However, in a practical consideration where a price needs to be paid for every kind of computation, a model is good if it can attain a high accuracy in as less epochs as possible. In such a situation, we find that fine-tuning ResNet 50 outperforms using Inception V3 as a feature extractor though the former process consumes more time and memory per epoch. We are thus able to see a superior performance of residual learning with a particular set of parameters. However, this accuracy is lost when we consider an increase in learning rate where the fine-tuning of the model shows poor results in the initial few epochs. Thus, though residual learning achieves a significant improvement, finding an optimal choice of parameters such as learning rate and number of epochs is an equally important factor

6.2 Need for More Data

In our work, we were restricted to using only the images provided by the Plant Village dataset and were unable to validate on other real-life examples since the data in sufficient numbers of different crops and diseases is not freely available. There are numerous works focusing on different databases. There is a urgent need to amalgamate all the databases in order to create a large verified database of such images. Further work in this area could look at collecting more images of leaves of these crops itself in different conditions as well as other crops and seeing if models as proposed by latest state-of-the-art techniques are able to generate good accuracy on it.

6.3 Practical Applications

We have built a very accurate model for detecting plant diseases which will be very useful for farmers lacking prior knowledge of crop diseases.

We have intentionally worked on a comparatively harder problem of detecting plant diseases assuming no information such as the name of crop is available which

for practical purposes can often be user provided. Our approach, however, is superior for cases when a fully automated maintenance system for multiple crops may be required.

In such a situation, our model will be able to correctly predict the plant and whether it is healthy or the disease it is suffering from. Such a use case may arise when case an unmanned aerial vehicle (UAV/ Drones) or unmanned ground vehicle (UGV), with a camera mounted on it is being used to make an automated assessment of a field. The camera mounted on these devices can take a photo from which leaves can be segmented and used to identify whether the crop is healthy or the specific disease it is suffering from.

6.4 Extensions

Further work in this area could look at integrating different kinds of information available to farmers, such as the time and place, and derivable attributes such as weather condition or any major local conditions to convert this problem to a multivariate problem. An ideal model will take the above-mentioned metadata along with images as an input and compute the disease and its severity and provide location or context-specific possible actions that a farmer should take.

References

1. Strange, R., Scott, P.R.: Plant disease: a threat to global food security. Phytopathology **43** (2005)
2. Tatem, A.J., Rogers, D.J., Hay, S.I.: Global transport networks and infectious disease spread. Adv. Parasitol. **62**, 293–343 (2006)
3. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D.: Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. (2016)
4. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? CoRR. arXiv:1411.1792 (2014)
5. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, pp. 248–255. IEEE (2009)
6. Huh, M.-Y., Agrawal, P., Efros, A.A.: What makes imagenet good for transfer learning? CoRR. arXiv:1608.08614 (2016)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR. arXiv:1512.03385 (2015)
8. Hughes, D.P., Salathé, M.: An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. CoRR. arXiv:1511.08060 (2015)
9. Amara, J., Bouaziz, B., Algergawy, A.: A deep learning-based approach for banana leaf diseases classification. In: BTW (Workshops), pp. 79–88 (2017)

10. Fuentes, A., Hyeok Im, D., Yoon, S., Sun Park, D.: Spectral analysis of CNN for tomato disease identification. In: International Conference on Artificial Intelligence and Soft Computing, pp. 40–51. Springer (2017)
11. Mwebaze, E., Owomugisha, G.: Machine learning for plant disease incidence and severity measurements from leaf images. In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 158–163. IEEE (2016)
12. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D.: Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. (2016)
13. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Front. Plant Sci. **7**, 1419 (2016)
14. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR. arXiv:1409.4842 (2014)
15. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. CoRR. arXiv:1512.00567 (2015)

Batch-Mode Active Learning-Based Superpixel Library Generation for Very High-Resolution Aerial Image Classification

Rajeswari Balasubramaniam, Srivalsan Namboodiri, Gorthi. R. K. Sai Subrahmanyam and Rama Rao Nidamanuri

Abstract In this paper, we introduce an active learning-based object training library generation for a multi-classifier object-oriented image analysis (OOIA) system. Given a sufficient number of training samples, supervised classification is the method of choice for image classification. However, this strategy becomes computationally expensive with an increase in the number of classes or the number of images to be classified. While several active learning approaches do exist for pixel-based training library generation and for hyperspectral image classification, there is no standard training library generation strategy for object-oriented image analysis (OOIA) of very high spatial resolution images. The above issue is solved in this proposed method where an optimised training library of objects (superpixels) is generated based on a batch-mode active learning (AL) approach. A softmax classifier is used as a detector in this method which helps in determining the right samples to be chosen for library updation. To this end, we construct a multi-classifier system with max-voting decision to classify an image at pixel level. This algorithm was applied on three different very high-resolution airborne datasets, each with varying complexity in terms of variations in geographical context, sensors, illumination and view angles. Our method has empirically outperformed the traditional OOIA by producing equivalent accuracy with a training library that is orders of magnitude smaller. The most distinctive ability of the algorithm is experienced in the most heterogeneous dataset where its performance in terms of accuracy is around twice the performance of the traditional method in the same situation.

Keywords Training library · OOIA · Active learning · Multi-classifier systems

R. Balasubramaniam · S. Namboodiri · G. R. K. S. Subrahmanyam (✉)
R. R. Nidamanuri
Indian Institute of Space Science and Technology, Thiruvananthapuram, India
e-mail: gorthisubrahmanyam@iist.ac.in; rkg@iittp.ac.in
URL: https://www.iist.ac.in/avionics/gorthisubrahmanyam

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_24

1 Introduction

With an exponential increase in the number of remote sensing satellites used since the past decade, an exorbitant amount of remote sensing data has been accumulated. With the advent of very high-resolution (VHR) technologies (QuickBird, WorldView-3, etc.), satellite and airborne remote sensing are currently undergoing a major technical revolution. High spatial resolution can provide extremely fine-grained details of the scene captured. This has tremendously increased the amount of information stored in these databases. These remote sensing data are extremely useful for numerous applications from change detection, landcover monitoring to GIS-ready information.

The traditional pixel-based classification methods have had the major drawback of lacking contextual information and high computational complexity. To satisfy the current requirements of classification of very high-resolution imagery, the recent trends have shifted towards object-oriented image analysis (OOIA) whereby an image is divided into a number of segments (objects) based on certain measures of similarity. This captures the local redundancies (reducing time complexity) and also measures the feature statistics over a naturally adaptive domain (object) [2].

As objects are the smallest units for classification, there cannot be one unique training library storing information as in the case of a spectral library. Hence, in OOIA, the training samples required to classify an image are taken from images of similar statistical characteristics which in common practice is the image itself. Although OOIA currently provides the most feasible solution for remote sensing image classification problem, obtaining the training samples is the most cumbersome process. Remote sensing image classification is a typical field wherein unlabelled data are available in abundance while labelling them requires considerable human effort and time. The problem with these classification methods is that they cannot use unlabelled data to train classifiers.

Active learning (AL) is a popular approach in machine learning that provides an alternative way to best represent a training library. AL iteratively selects the most informative samples to be labelled. AL procedures are widely used in a variety of fields, including medical imaging, biometrics [3], computer vision image segmentation tasks. AL method is widely formulated in pixel-based classification of hyperspectral images using spatial and spectral information [5]. Most of the classification strategies work at pixel level, and this becomes a problem when the work has to be extended to VHR imagery.

The work which is closest to the one proposed in this paper is in the domain of hyperspectral image classification in [6].

In this paper, we propose a novel active learning methodology to form a compact generic discriminative dictionary of training objects (superpixels) from a large database of heterogeneous images, in order to produce high classification accuracy with a very small set of labelled samples.

This is achieved through the following sub-contributions:

- Identification of best representative image(s) from the database and quickshift superpixelisation of the same to form an initial object training library.

- Updating the library with the help of the decisions of a supervised classifier (detector) used in an unsupervised methodology by using active learning strategies.
- Utilisation of the generated library in a multi-classifier OOIA system.
- Extensive evaluation of the proposed methodology and demonstration of its capabilities over three interesting datasets. Implementation of the same approach to multi-spectral images and cross-domain applications.
- Comparison against self-learning-based semi-supervised SVM approach for classification of such heterogeneous data.
  The datasets are chosen specifically to test its robustness with respect to variation in geographical context, view angles, sensors and other parameters encompassing all the images present in the database.

The rest of this paper is organised as follows. Section 2 describes the proposed methodology and the algorithms involved. Section 3 gives an insight into the experiments performed, evaluation of results and discussions related to certain interesting observations. In Sect. 3.4, we describe the performance obtained over various datasets. The final section concludes with discussions on plausible variations for further development of the algorithm to improve its performance.

2 Proposed Methodology

The active learning methodology proposed here constitutes of four major stages, namely representative image generation stage, object generation and initial library generation stage, detection and updation stage and finally the multi-classifier stage. The detailed description about each of these stages is described in the following subsections.

2.1 Representative Image Generation Stage

Initial library generation is an important stage of the entire process due to the dependence of the detector on the initial library. If the detector performance is deteriorated, an optimised training library cannot be obtained. Hence, for an ideal performance of the detector, we must stress on the right choice of an initial set of images for the training library. Basically, the initial objects in the training library should be the best representation of the entire dataset. To accomplish this, identifying the best representative image(s) is necessary. It can be assumed that the image containing the maximum information content will contain objects with high degree of inter-class and intra-class variance. Hence, an initial unbiased sampling of objects can be achieved. We perform principle component analysis (PCA) over all the images in the database and obtain the entropy of the principle component of each of the image as a measure of its information content. This is shown as the first step in Fig. 1.

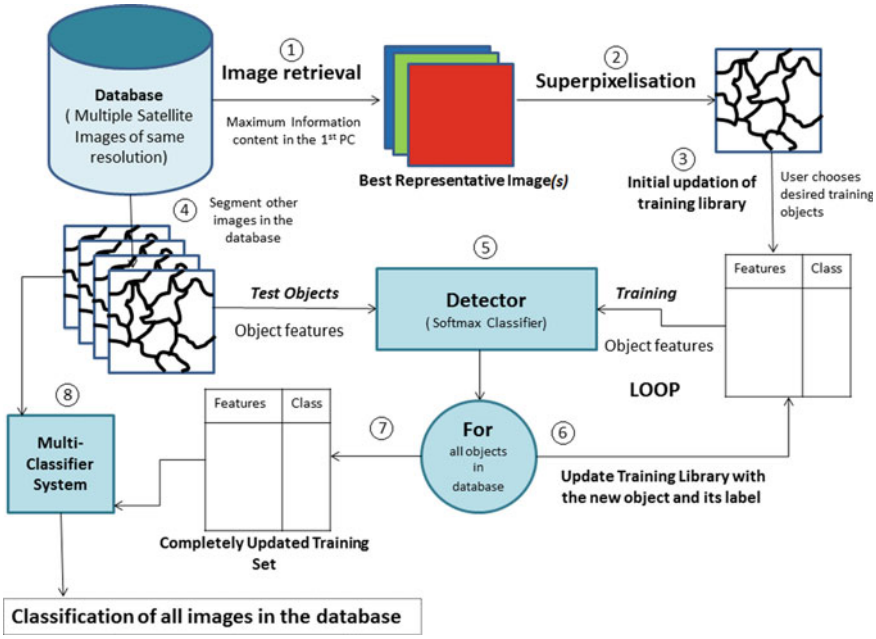


Fig. 1 Active learning block diagram representation

For instance, let us consider images with three bands (RGB). Then, each pixel is a 3D random variable. To obtain the principle component of the image, we project all the pixels in the direction of maximum variance.

$$\psi = \frac{1}{m * n} \sum_i \Gamma_i \tag{1}$$

where ψ is a three-dimensional mean vector over the bands,
m and n denote the spatial extent of any given image (total number of pixels),
i refers to the pixel count in the given image, $(i = m * n)$, and
Γ_i is the ith pixel vector.

$$C = \frac{1}{m * n} \sum_i (\Gamma_i - \psi)(\Gamma_i - \psi)^T \tag{2}$$

C is the scatter matrix of size 3×3. Let U_1, U_2, and U_3 be the eigen vectors arranged according to the descending order of their corresponding eigen values. Therefore, the principle direction of variance is along U_1.

To obtain the principle component image:

$$\omega_i = U_1{}^T(\Gamma_i - \psi) \tag{3}$$

where ω_i represent the generated principle components.
To evaluate the information content over the PCs, we measure its entropy. Entropy for a set of random variables is given by:

$$E = -\sum_i (\omega_i * log_2(\omega_i)) \tag{4}$$

Once the entropy is found for all the images, the images are arranged in the descending order of their entropy. The top K images represent the K best representative images. Now, any number of images can be chosen as per the entropy order until equal objects are generated per class in the next stage.

2.2 Object Generation and Initial Library Generation Stage

Once the best representative images are identified, the next step is the initial library generation. For this, we need objects from the best representative images. These objects are generated by the means of quickshift superpixelisation scheme. Quickshift algorithm can produce segments without the user specifying the number of segments required and is predominantly mode seeking over intensity and spatial vectors [1] which is ideal for satellite image segmentation. Quickshift is a mode-seeking algorithm with a time complexity of $\mathcal{O}(dN^2)$ [9]. The superpixelisation is pictorially represented in Step 2 of Fig. 1. Now, while labelling the objects, care must be taken not to make the training database unbalanced. The initial library generation is shown in Step 3 of Fig. 1. The same superpixelisation scheme is applied over all the images in the database to generate the test database for classification. This is shown as Step 4 in the same figure.

2.3 Detection and Updation Stage

Detection is performed with the help of the softmax classifier. The prediction of the classifier is used to identify objects that have not been witnessed by the library before. From this ambiguous object space, only those samples with an area greater than a limiting area threshold are chosen for updation into the training library. Features are extracted from these newly observed objects and are updated into the training library with their appropriate classes. The training of the softmax classifier refers to Step 5 in Fig. 1. The features extracted can be broadly split into intensity-based features and texture-based features. Intensity-based features include sample mean and standard

deviation across the superpixel. Entropy is a statistical measure of randomness that can be used to characterise the texture of the input region.

$$\mu_j = \frac{\sum_i x_{ij}}{N} \tag{5}$$

$$\sigma_j = \frac{\sum_i (x_{ij} - \mu_j)^2}{N} \tag{6}$$

$$\xi = -\sum (p. * log_2(p)) \tag{7}$$

where

x_i pixels present in the given superpixel.
N total no. of pixels in the given superpixel.
μ_j mean pixel intensity in band j.
σ_j Std. deviation of intensity in band j.
ξ entropy.
p histogram counts of intensity values over all bands.

Hence, in our study, the feature vector over every object is a $\Re^7$ vector containing the intensity and texture information. This training library is then fed into a softmax regressor.

Softmax regression is a generalisation of logistic regression for multiple classes. The formulation of softmax regression has been obtained from [8]:

Once the softmax classifier assigns posterior probabilities within the ambiguous range, i.e. around 0.5, where uncertainty is high for each of the classes, those objects are considered as ambiguous and are assumed to have not been witnessed by the training library before. Hence, these objects are updated into the training library after an area filter with corresponding labels as shown in Step 6 of Fig. 1.

2.4 Multi-classifier Stage

The multi-classifier system (MCS) that we have designed contains five classifier modules, namely one-vs-all RBF-SVM, binary decision tree, AdaBoost classifier, K-nearest neighbours and Gaussian discriminant analyser. The motive of choosing such wide variety of classifiers is that each classifier as an individual has its own special capabilities. We are interested in utilising the varied abilities of all these classifiers to arrive at a plausible solution. The decision fusion in this case is performed by max-voting technique. An important point of interest is that the detection and classification are separately handled in the proposed algorithm unlike in typical AL methods.

3 Experiments, Results and Analysis

3.1 Location and Datasets

The proposed method was implemented, and its performance has been evaluated for the classification of various high-resolution aerial three band and multi-spectral imagery. The dataset consists of multi-sensor images that reflects different spatial resolutions and geographic locations which encompass different types of landcover settings.

We have chosen three RGB datasets that will satisfy all the above requirements. ISPRS segmentation dataset [4] contains adjacent image patches from a single satellite swath. These images belong to the same geographical location which is Vaihingen, Germany. The Massachusetts road dataset [7] is publicly made available by the University of Toronto. This dataset contains independent aerial images captured in the state of Massachusetts. The third dataset is provided by the University of California, Merced [10]. This dataset contains a wide range of images from varying geographical context across the USA, captured through a wide variety of sensors and varying viewpoints. Evaluation over this dataset will provide the actual worst-case performance of the proposed methodology.

3.2 RGB—Urban Domain Analysis

Dataset 1 describes the best-case scenario and Dataset 3 describes the worst-case scenario. Comparisons are made based on these two cases. The average number of objects formed for Dataset 1 were 43,237 objects. For Dataset 3, the average number of objects formed were 36,282. We have utilised just 100 objects of this total number of objects to achieve 85.22% accuracy in Dataset 1 and 71.61% in Dataset 3. To achieve the similar range of accuracy by the conventional choice of objects, we would need 5200 objects in Dataset 1 and 2000 objects in Dataset 3. This means we have reduced the total number of training samples by 52 times in Dataset 1 and reduced the number of training objects by 20 times in Dataset 3. We are able to achieve this optimisation due to the detector mechanism that helps choose the most ideal training objects and reduce the otherwise prominent redundancy. Table 1 consolidates the above analysis. From the sample image classification results shown in Figs. 3 and 2, it can be visually verified that the four classes: roads (cyan), vegetation (blue), buildings (yellow) and shadow regions (red) have been distinguished effectively.

Table 1 Comparison of best-case and worst-case scenario for the proposed and conventional methodology

Dataset description	Total objects in the database	Classification accuracy (%) with 100 optimised samples	Samples required by conventional method for same accuracy	Factor reduction
Best-case scenario (Dataset 1)	43,237	85.22	5200	52 times
Worst-case scenario (Dataset 3)	36,282	71.61	2000	20 times

Table 2 Cross-domain evaluation results

Dataset	Domains	Classes	No. of objects	Training samples (%)	Classification accuracy (%)	Computation time (min)
UC merced 500 images	5	9	1,22,466	~3	55.69	8
				~5	63.34	12

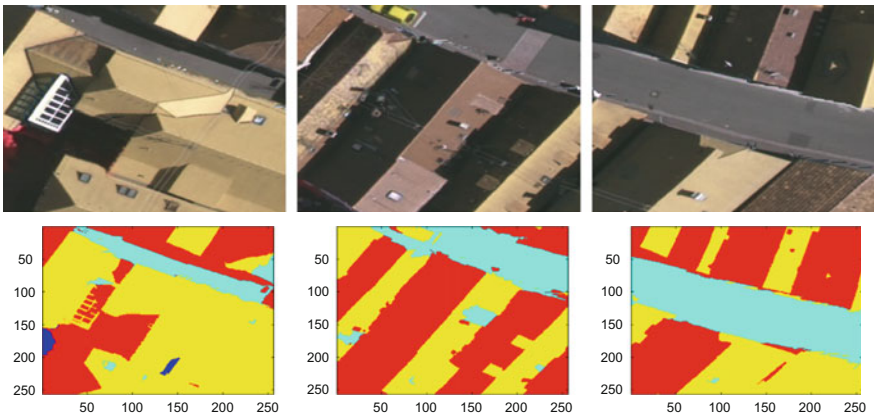


Fig. 2 Classification result for 113 updations in Dataset 1 (best-case scenario)

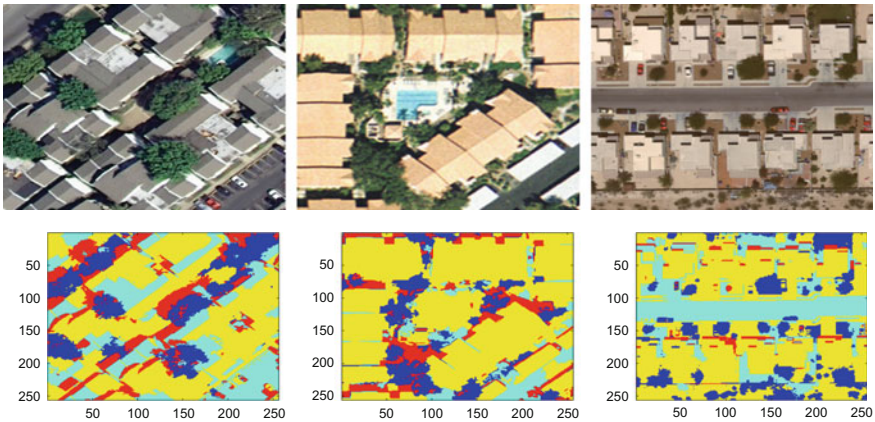


Fig. 3 Classification results for 207 updations in Dataset 3 (worst-case scenario)

3.3 RGB—Cross-Domain Analysis

In the previous sections, the performance of the algorithm is tested over a single domain, namely urban. In this section, we move one step further by testing the performance of the proposed algorithm over a database consisting of images from five different domains and a total of nine different classes. The domains used are urban, forest, agriculture, harbour and coastal. The nine different classes include sand shore, sea, buildings, roads, wild trees, ships, barren land, crops and background. The images are obtained from the UC Merced Landuse dataset. UC Merced dataset provides 21 different domains with 100 images per domain. As this evaluation utilises 5 random domains, therefore 500 images were used for this purpose. The classification is performed for two different training sample sizes, 3 and 5% with validation done over number of samples equal to the size of training samples. It can be seen from Table 2 that there is a considerable improvement in classification accuracy for a small increase in training samples. However, the performance is not as high as in the case of single domains. It is worthwhile to note from the accuracy of Dataset 3 of Table 5 that there is only a slight drop in the performance of the algorithm though the complexity of the database has increased by several folds.

Table 3 Traditional classification strategy

Datasets	Accuracy for a single image (%)	
	5 samples per class	20 samples per class
Dataset 1	85.65	96.84
Dataset 2	75.24	91.82
Dataset 3	71.66	84.32

Table 4 Evaluation of number of samples required for similar range of accuracies

Datasets	Our method no. of samples	Accuracy (%) achieved with the specified no. of samples in our method	Traditional method no. of samples	Accuracy (%) 20 samples/class traditional method
Dataset 1	921	95.37	20,800	96.84
Dataset 2	423	90.3	16,000	91.82
Dataset 3	1124	83.55	8000	84.32

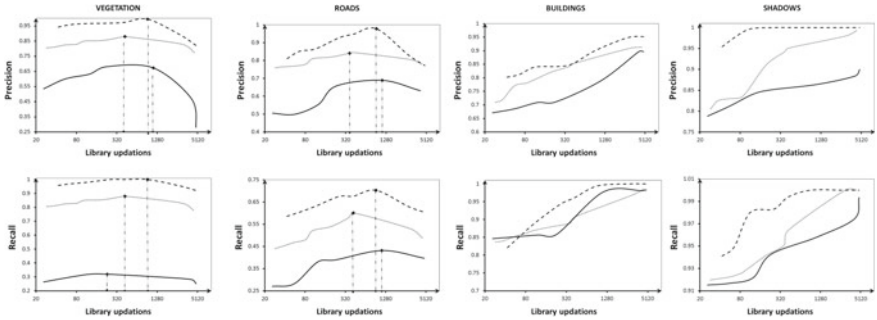


Fig. 4 Variation of performance parameters: precision and recall for individual classes with library updations

3.4 Performance Comparisons

First, the proposed methodology is tested against the traditional method. Table 3 evaluates accuracy over a single image when only 5 samples are taken per class and when 20 samples are taken per class. The classes considered in our case include (1) vegetation, (2) roads, (3) buildings and (4) shadow regions. Therefore, in the five samples per class case, each image needs $(5 * 4) = 20$ objects for a given image. It can be seen from the tabulation that for Dataset 1 to achieve 85.65% roughly over all the images, it needs $(20 * 260) = 5200$ objects. For Dataset 2, to achieve 75.24% it needs $(20 * 200) = 4000$ objects, and for Dataset 3, to achieve 71.66% accuracy, $(20 * 100) = 2000$ objects are necessary. Similarly on the higher end, for the 20 samples per class case, 20,800, 16,000 and 8000 objects are required for Datasets 1, 2 and 3 to achieve their corresponding accuracies. In Table 4, the number of objects required for achieving a similar range of accuracy is compared between our method and the traditional method. It is interesting to note that, above the specified number of objects in our method, the accuracy begins to saturate and only a small increase in the accuracy is observed with further updations in all the three datasets.

Figure 4 depicts the extensive evaluation of the effect of library updation over the performance parameters—accuracy, precision and recall. Certain interesting obser-

Table 5 Comparison of performance with and without library updations

Dataset	Accuracy (%) with 100 training samples	
	With library updations	Without library updations
Dataset 1	85.22	85.13
Dataset 2	84.81	81.13
Dataset 3	71.61	35.11

vations can be noted from the variation of the performance parameters with the library updations. Firstly, all the plots seem to saturate after a certain number of training library updations; i.e. the gradient of these variables reduces with increasing library updations. The most important observation is that, precision and recall of vegetation and road classes droop down after a certain amount of library updations. The droop points are shown in their corresponding plots in Fig. 4. This can be attributed to the fact that in the detection stage, the objects that have not been seen by the library before are updated into the library. In the due course of this action, those classes which have high variation through all the images in the given database get the maximum updation into the training library and hence an unbalanced situation occurs. In our datasets, maximum variation occurs in the building class and also in the shadow class; variation in the latter case is intuitive. Vegetation and roads are more or less uniform in terms of within-class variation. Hence, the precision and recall of vegetation and road classes reduce after a certain number of updations due to the imbalance in the training data. However, this shows no impact on the accuracy of overall classification.

The most important comparison lies between the performance of classification with and without library updations. This is shown in Table 5. It can be seen that for just 100 optimised samples in the final training library, there is an improvement in the classification accuracy among all datasets as against without updations. Noteworthy improvement is observed in its performance in Dataset 3. The classification accuracy has improved from 35.11 to 71.61%. Dataset 3 has the maximum variation in terms of geographical context, viewpoints, etc. The traditional method fails absolutely in this case. This is the most important contribution of the proposed method.

4 Conclusion

The analysis described in this paper suggests that a compact generic discriminative dictionary of training objects from a large database of heterogeneous images is sufficient to produce high degree of accuracy during the classification of the database. In the traditional methods of classification, the training objects were highly custom-typed to the image that has to be classified. It was found after detailed analysis that a small set of highly distinct training objects can produce better performance than

a large training database with redundant objects and less information content. The presented method has also displayed its generality and robustness by showing its capability over cross-domain data.

Acknowledgements The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [4] http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Blaschke, T., Lang, S., Hay, G.: Object-based image analysis: spatial concepts for knowledge-driven remote sensing applications. Springer Science & Business Media (2008)
3. Chakraborty, S., Balasubramanian, V., Panchanathan, S.: Adaptive batch mode active learning. IEEE Trans. Neural Netw. Learn. Syst. **26**(8), 1747–1760 (2015)
4. Cramer, M.: The DGPF-test on digital airborne camera evaluation—overview and test design. Photogrammetrie-Fernerkundung-Geoinformation (2), 73–82 (2010)
5. Ding, S., Li, B., Fu, X.: Active learning methods for classification of hyperspectral remote sensing image. In: International Conference on Intelligent Computing. pp. 484–491. Springer (2014)
6. Guo, J., Zhou, X., Li, J., Plaza, A., Prasad, S.: Superpixel-based active learning and online feature importance learning for hyperspectral image analysis. IEEE J. Select. Topics Appl. Earth Observe. Rem. Sens. (2016)
7. Mnih, V.: Machine learning for aerial image labeling. Ph.D. thesis, University of Toronto (2013)
8. Ng, A., Ngiam, J., Foo, C.Y., Mai, Y., Suen, C.: UFLDL tutorial. http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial (2013)
9. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Computer Vision—ECCV 2008, pp. 705–718. Springer (2008)
10. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 270–279. ACM (2010)

An Efficient Algorithm for Image Haze Removal in Outdoor Environment

Himanshu Dua, Teena Sharma, Pooja Agrawal and Nishchal K. Verma

Abstract Computer vision applications in outdoor environment are mainly affected by factors such as pollution, clouds, shadow, haze, fog. Herein, an algorithm for real-time haze detection and removal in image has been implemented. For this purpose, the Dark Channel Prior technique is used which is efficient method for haze removal and also contains information about the level of haze in the image. Initially, visibility index of image frame is estimated to determine whether image is hazy or not using haze detection model. Thereafter, hazy images are processed with haze removal model to enhance the image visibility. The haze removal model uses guided filter to accelerate the process. Non-hazy images do not require to process with haze removal model. The algorithm has been tested for four datasets, i.e., non-hazy, slightly hazy, medium hazy, and heavily hazy. The proposed algorithm is performing well and also able to tackle the halo effects at some extent.

Keywords Haze detection · Haze removal · Dark Channel Prior · DCP · Dark Channel Ratio · DCR · Visibility index · Guided filter

T. Sharma (✉) · P. Agrawal · N. K. Verma
Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: tee.shar6@gmail.com

P. Agrawal
e-mail: pooja.iitd@gmail.com

N. K. Verma
e-mail: nishchal.iitk@gmail.com

H. Dua
Indian Institute of Technology Roorkee, Roorkee 247667, India
e-mail: himanshudua.uec2015@iitr.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_25

1 Introduction

Computer vision enables a machine or computer to see and visually sense the world around them. It is related with the automatic extraction, analysis, and better understanding of particular information from a single image or a sequence of images. Some of its useful applications includes image retrieval [1–3], image classification [4–7], image analysis and recognition [8–10], aerial Imagery [11], remote sensing [12–14], future image frame prediction [15–17]. The most important factor, on which computer vision depends, is the quality of an image for further processing in various fields. The light, pollution, clouds, shadow, haze, fog, etc., are some factors in outdoor environment which affect the quality and accuracy of astronomical observations and images. Turbulence and temperature variations in the Earth's atmosphere cause astronomical objects to appear to twinkle and form blurry images. Therefore, the visibility enhancement approaches such as haze removal can contribute toward vision-based applications to deal with outdoor factors by enhancement of image quality.

In the literature, the traditional methods used for haze removal in input image were mainly depend on histogram-based dehazing methods. These methods include contrast enhancement [18], adaptive contrast enhancement [19], advanced contrast enhancement [20] using histogram equalization. The traditional methods were able to enhance the degraded quality of image due to haze in a single image frame. In recent work, the main focus is to improve the performance in multiple images for real-time applications such as tracking, recognition in outdoor environment. The polarization-based techniques [21–23] were used for this purpose. The polarization-based techniques are rely on the fact that scattered air light due to atmospheric particles is partially polarized as image is composed of scene radiance and air light. The polarization-based filtered images can also be used for specularity analysis [24–26], semi-reflected scenes [27–29], material classification [30], scene segmentation [31], etc. The real-time polarization information can also be extracted through polarimetric cameras [32]. In [33–35], Narasimhan et al. discussed approaches for image dehazing under different weather conditions for multiple images. Image dehazing has also been done using depth information in [36, 37]. The image information via color, texture, lines majorly affects due to degraded quality of image in outdoor environments. However, indoor applications of identification using these attributes have been discussed in [38–44].

In this paper, an algorithm for haze detection and removal in image frames has been proposed. Herein, Dark Channel Prior (DCP) method, haze detection model, haze removal model, and guided filter are discussed for implementation of proposed algorithm. The first step is to find out the presence of haze in an image using DCP method in haze detection model which is also termed as visibility index. If haze is present in the image, the image is passed to haze removal model for image enhancement else non-hazy image is directly passed for further processing. If haze is present, next step is to remove the haze such that information content of image must not vary. These information contents of an image, i.e., edges, lines, texture, are responsible to

extract out the image features for vision-based applications. The results are compared with state-of-the-art method discussed in [45].

The rest of the paper is arranged as follows: Sect. 2 highlights the visibility index calculation method. Section 3 discussed the step-by-step implementation of proposed methodology. Section 4 shows the comparison of results with state-of-the-art method. Finally, Sect. 5 discussed the conclusions of the paper.

2 Dark Channel Prior

As shown in [45], Dark Channel Prior is an effective method for removing haze from the image. In computer vision and computer graphic, the most widely used model to describe the formation of a hazy image is the atmospheric scattering model. According to this model, the captured image can be modeled as two components: *the direct reflection of light from the object* and *the reflection from the particles of the medium*, it is defined as follows:

$$\mathbf{I}(x) = \mathbf{J}(x)t(x) + \mathbf{A}(1 - t(x)) \tag{1}$$

where $\mathbf{I}$ is the observed intensity, $\mathbf{J}$ is the scene radiance, $\mathbf{A}$ is the global atmospheric light, and t is the medium transmission describing the portion of light that is not scattered and reaches to the camera. In case of homogeneous atmosphere, t is expressed as:

$$t(x) = e^{-\beta d(x)} \tag{2}$$

where β is the scattering coefficient of the atmosphere and d is the scene depth. The equation reveals the relationship between scene depth and medium transmission. It shows that the scene radiance $\mathbf{J}$ exponentially attenuated with the scene depth.

A very nice use of the model for haze removal is done through the Dark Channel Prior in [45]. As the images captured in nature are always colorful and full of shadows, the Dark Channel Prior is valid in most of the cases.

The Dark Channel J for a RGB image is defined as:

$$J^{\text{dark}}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in r,g,b} J^c(y) \right) \tag{3}$$

where J^c is one of the color channels of input RGB image and $\Omega(x)$ is a local patch centered at x. The minimum filter image and the Dark Channel image for an example hazy image are shown in Fig. 1.

Now, in accordance with the above-mentioned observation, if input image is a haze-free image, then

$$J^{\text{dark}} \rightarrow 0 \tag{4}$$

(a) Original Image (b) Output after applying Minimum Filter (c) Dark Channel of the image

Fig. 1 Minimum filter and Dark Channel for an example image

Now, if the atmosphere light **A** is known, and transmission $t(x)$ is assumed to be constant ($t^{estimate}$) in a local patch $\Omega(x)$ as assumed in [45], Eq. 3 can be rewritten as:

$$\frac{I^c(x)}{A^c} = t^{\text{estimate}}(x)\frac{J^c(x)}{A^c} + 1 - t^{\text{estimate}}(x) \tag{5}$$

Now, taking Dark Channel on both and applying the above-mentioned approximation in Eq. 4, the transmission can be calculated as:

$$t^{\text{estimate}}(x) = 1 - \min_{y\in\Omega(x)}\left(\min_{c\in r,g,b}\left(\frac{I^c(y)}{A^c}\right)\right) \tag{6}$$

However, atmosphere is not absolutely free of any particle even in clear days. So there is a little haze existent for distant objects, and the haze is a fundamental cue for human to perceive depth [46]. We call this phenomenon aerial perspective. So a constant parameter ω which is an application-based value is introduced into Eq. 6 to keep a very small amount of haze for the distant objects, making the haze-free image seem natural.

$$t^{estimate}(x) = 1 - \omega \min_{y\in\Omega(x)}\left(\min_{c\in r,g,b}\left(\frac{I^c(y)}{A^c}\right)\right) \tag{7}$$

For calculation of better transmission, He et al. [45] used soft matting or Laplacian matting matrix [45] to preserve the edges and refine the transmission value as well. However, this process is very time consuming and should be replaced with something faster for making real-time execution of algorithm possible. This is discussed in Sect. 3.2.

3 Proposed Methodology

3.1 Haze Detection Model

The method that we propose here uses the information from the Dark Channel Prior of input image to calculate a visibility index for the image, namely Dark Channel Ratio (DCR) [47]. The flowchart for the model is shown in Fig. 2. This is described with detailed explanation in the following subsections.

Dark Channel Ratio Visibility index for an image is a term describing the clarity and sharpness of the image contents. This can also be defined as a way of describing the density of haze in the image as the factors that decrease the visibility of the image, such as blurriness, are somewhat related to the haze in the image (or could be considered similar to haze for computer vision purposes). Therefore, Dark Channel prior (described in next section), a statistical model describing the haziness of an image, is used here to define a visibility index for images. As described in [47], Dark Channel Ratio (DCR) is defined as the ratio of the number of pixels in the Region of Interest (ROI) (K) with intensity less than a threshold value to the total number of pixels in the ROI of the image.

$$DCR = \frac{K}{M \times N} \tag{8}$$

where M and N denote the width and length of the ROI of the image, respectively. Originally, the ROI is defined as the whole image except for the 15% pixels in the top as well as in the bottom of the image (in order to attenuate disturbances caused by sky and road regions). We use here the same approach.

The lesser the DCR is, the more is the haze in the image. Therefore, we have set a threshold value for the DCR here. All the images having DCR less than the threshold value are considered as hazy and are processed further to remove the haze from them. On the other hand, the images with DCR greater than threshold are considered as haze-free and no processing for haze removal is done on them.

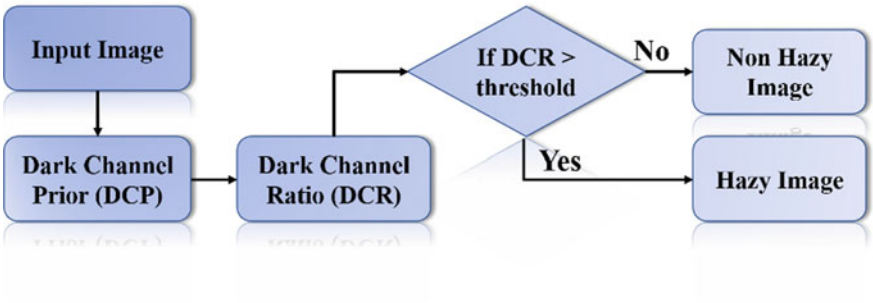


Fig. 2 Flowchart for the Haze detection model

Selecting Threshold Value: Logistic Regression Now that we have defined a parameter for the detection of haze, we need to select a threshold value for the parameter such that any DCR value below the threshold means the image is non-hazy, and vice versa. For this purpose, we have used simple logistic regression. Logistic regression is a simple, yet elegant method for solving any classification problem. Its most simple version, known as simple logistic regression, is used when we have one nominal variable and one measurement variable. This method finds the best equation that could predict the value of the nominal variable (Y) for every measurement variable (X). Although it is quite similar to linear regression, it is different in the fact that rather than measuring the variable Y directly, it gives a probability of obtaining a certain value of Y. The logistic regression equation is defined as follows:

$$ln[\frac{Y}{1-Y}] = a + bX \tag{9}$$

where the slope (b) and intercept (a) of the equation are calculated using the maximum likelihood method. It means that the values are calculated such that we get as close as possible to the observed results. We have used gradient descent [48] here to calculate the results.

We applied the logistic regression on our dataset and then calculated the threshold value by putting the regression output as 0.6, denoting the value with 0.6 probability of image being hazy. The probability is selected as more than 0.5 to provide better results.

3.2 Haze Removal Model

Once the image is detected as hazy, it is processed using the already calculated Dark Channel of the image. The flowchart for complete process is shown in Fig. 3. The haze-free image is calculated using the method of Dark Channel Prior as described in Sect. 2.

However, the above defined method includes soft matting for the refinement of transmission, which is very time consuming and makes it impossible to deal with real-time images. Therefore, to make the process faster, He et al. [49] later came up with a new, faster technique called guided filter. This is explained in the next subsection.

In the earlier work [45], ω was taken to be 0.95. However, we choose ω here to be 0.9 as it gave better results during the experiments. Moreover, the patch sizes were also kept same earlier. However, it was observed that the results were better when the patch size was decreased while estimating the value of t. The smaller patch size led to decrease in the halo effects of the image. With larger patch size, the transmission value was over-estimated and thus, more halo effects were observed. Therefore, in order to avoid over-estimation of t, a separate, small patch size is used for the calculation of Dark Channel during estimation of t, other than the one used

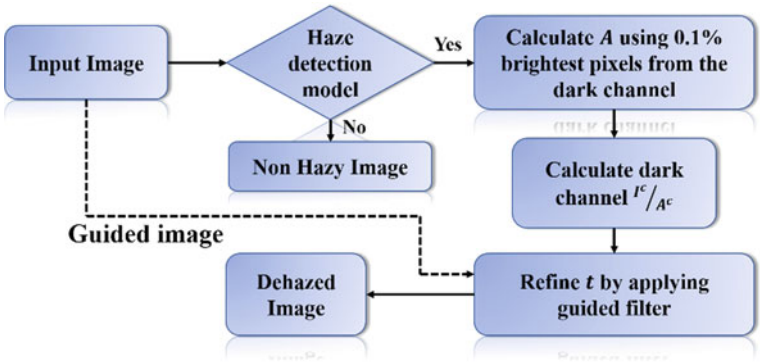


Fig. 3 Flowchart for the Haze removal model

for the calculation of Dark Channel of the original image. As a result, the halo effects become somewhat less as the over-estimation has decreased.

Guided Filter Guided filter is a technique to utilize the features of an guidance image in order to enhance the input image. Moreover, the theoretical aspects of the guided filter are connected with the Laplacian Matrix of matting (the method used earlier for transmission estimation), which suggests that it can be used for the refinement of transmission as well. It is defined as a local linear model, which is given below:

$$t_i = a_k I_i + b_k, \forall i \in \omega_k \tag{10}$$

where a_k and b_k are linear coefficients and are constant for window ω_k centered at pixel k. The hazy image I is used as the guidance image here, while transmission t is the final output here.

A cost function E, similar to as defined in linear regression, is minimized here as we decide the values of a_k and b_k. It is defined as follows:

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - t_i^{estimate})^2 + \epsilon a_k) \tag{11}$$

The new term defined here, ϵ, is to regularize the value of a_k. The values of linear coefficients are then derived by applying the linear regression model:

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i t_i^{\text{estimate}} - \mu_k \bar{t}_i^{\text{estimate}}}{\sigma_k^2 + \epsilon} \tag{12}$$

$$b_k = \bar{t}_i^{\text{estimate}} - a_k \mu_k \tag{13}$$

Although the defined algorithm is for a gray image, it can easily be applied on color images by simply applying it on all three color domains (RGB) of the image.

4 Experimental Results

The methodology that we have proposed here is executed through MATLAB and is carried out on a PC with Intel core i7 and 8 GB RAM. As already mentioned, soft matting or the Laplacian matrix is a very time-consuming method for refinement of transmission and is therefore not suitable for real-time execution of haze removal. Therefore, guided filter is suited for the process. The Dark Channel Prior method was applied on 16 different images using soft matting and guided filter in the refinement step. The data for comparison between calculation time of both methods are shown in Fig. 4. Moreover, the results for both methods looked almost similar, although the edges were slightly sharper in case of guided filter. The results for some example images are shown in Figs. 5, 6, 7, and 8.

We have applied the method mentioned above on images that are divided into four datasets. Each dataset has ten images with the level of haze common for all of them, i.e., not hazy, slightly hazy, medium hazy, and heavily hazy. The images for these datasets are shown in Fig. 9. The data for the DCRs of these datasets are given in Table 1. While applying our method, the DCR for all of the datasets was calculated in order to calculate the threshold value of DCR for haze detection. Once the DCRs were calculated, we applied logistic regression on them to find out the threshold value. By applying the chi-square test for 95% confidence intervals, we get the regression equation coefficients $a = -3.3854$ and $b = 11.1236$. Putting $Y = 0.6$ along with the coefficients in the equation, we get $X = 0.341$. Therefore, we select threshold value for DCR as 0.341.

While processing the images for removing haze, the different parameter values were tested for parameters like ω, patch size in Dark Channel, and neighborhood size in guided filter. As already mentioned, the value of ω was varied and it was found out

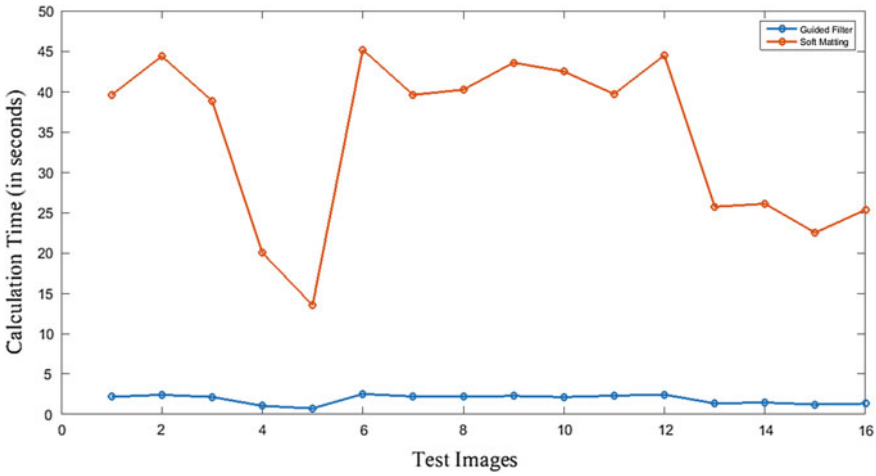


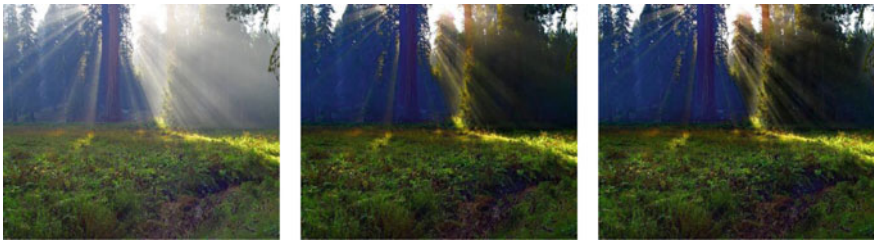
Fig. 4 Graph for soft matting and guided filter calculation time

(a) Original hazy image (b) Output using Soft Matting (c) Output using Guided Filter

Fig. 5 Test image 1—comparison of results with soft matting and guided filter

(a) Original hazy image (b) Output using Soft Matting (c) Output using Guided Filter

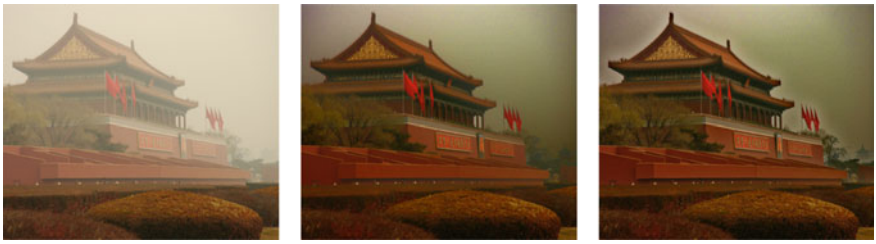
Fig. 6 Test image 2—comparison of results with soft matting and guided filter



(a) Original hazy image (b) Output using Soft Matting (c) Output using Guided Filter

Fig. 7 Test image 3—comparison of results with soft matting and guided filter

that lowering its value a bit helped in improving the algorithm a bit, in the sky regions where it shows some problems in clearing the image. Hence, we have selected the value of ω as 0.90. Moreover, again as already discussed, the tuning of the patch size resulted in the fact that when a smaller patch size is selected during the estimation of transmission, the halo effects are reduced and the image becomes clearer with sharper edges. Thus, while the patch size for Dark Channel during calculation of A is 15 as in the previous works, during estimation of transmission, it is selected as



(a) Original hazy image (b) Output using Soft Matting (c) Output using Guided Filter

Fig. 8 Test image 4—comparison of results with soft matting and guided filter

(a) Dataset 1: Non Hazy (b) Dataset 2: Slightly Hazy (c) Dataset 3: Medium Hazy (d) Dataset 4: Heavily Hazy

Fig. 9 Dataset images

Table 1 Values of DCR for different datasets

Dataset 1		Dataset 2		Dataset 3		Dataset 4	
Image	DCR	Image	DCR	Image	DCR	Image	DCR
1	0.252	1	0.005	1	0	1	0.009
2	0.174	2	0.043	2	0	2	0
3	0.337	3	0.003	3	0.128	3	0.193
4	0.656	4	0	4	0.135	4	0
5	0.800	5	0.638	5	0.104	5	0
6	0.317	6	0	6	0.0308	6	0.006
7	0.563	7	0	7	0	7	0
8	0.444	8	0.102	8	0	8	0.102
9	0.527	9	0.032	9	0.002	9	0.001
10	0.967	10	0.137	10	0.012	10	0

3. In guided filter, the neighborhood size, initially set at 5, was increased until the images became almost completely haze-free and was set to 32. Increasing the size further did not make much difference to the output.

5 Conclusions

The proposed algorithm performs well in outdoor environment for haze removal with less computational complexity in comparison of state-of-the-art methods. The heuristics selection with respect to size of input image makes the calculation faster. The algorithm initially identifies that the image is hazy or non-hazy which speeds up the algorithm by processing input image via haze removal model only when the image is hazy. The algorithm is also able to deal with halo effects at some extent during haze removal. The future work can be done by improving the performance of algorithm using depth data.

References

1. Han, J., Ji, X., Hu, X., Zhu, D., Li, K., Jiang., X., Cui, G., Guo, L., Liu, T.: Representing and retrieving video shots in human-centric brain imaging space. IEEE Trans. Image Process. **22**(7), 2723–2736 (2013)
2. Han, J., Ngan, K., Li, M., Zhang, H.-J.: A memory learning framework for effective image retrieval. IEEE Trans. Image Process. **14**(4), 511–524 (2005)
3. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **28**(7), 1088–1099 (2006)
4. Shao, L., Liu, L., Li, X.: Feature learning for image classification via multi objective genetic programming. IEEE Trans. Neural Netw. Learn. Syst. **25**(7), 1359–1371 (2014)
5. Zhu, F., Shao, L.: Weakly-supervised cross-domain dictionary learning for visual recognition. Int. J. Comput. Vis. **109**(12), 42–59 (2014)
6. Luo, Y., Liu, T., Tao, D., Xu, C.: Decomposition-based transfer distance metric learning for image classification. IEEE Trans. Image Process. **23**(9), 3789–3801 (2014)
7. Tao, D., Li, X., Wu, X., Maybank, S.J.: Geometric mean for subspace selection. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 260–274 (2009)
8. Liu, L., Shao, L.: Learning discriminative representations from RGB-D video data. In: Proceedings International Joint Conference Artificial Intelligence, pp. 1493–1500. Beijing, China (2013)
9. Tao, D., Li, X., Wu, X., Maybank, S.J.: General tensor discriminant analysis and Gabor features for gait recognition. IEEE Trans. Pattern Anal. Mach. Intell. **29**(10), 1700–1715 (2007)
10. Zhang, Z., Tao, D.: Slow feature analysis for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 436–450 (2012)
11. Woodell, G.A., Jobson, D.J., Rahman, Z.U., Hines, G.: Advanced image processing of aerial imagery. Proc. SPIE**6246**, 62460E (2006)
12. Han, J., Zhang, D., Cheng, G., Guo, L., Ren, J.: Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. IEEE Trans. Geosci. Remote Sens. **53**(6), 3325–3337 (2015)
13. Cheng, G., Han, J., Guo, L., Qian, X., Zhou, P., Yao, X., Hu, X.: Object detection in remote sensing imagery using a discriminatively trained mixture model. ISPRS J. Photogramm. Remote Sens. **85**, 32–43 (2013)
14. Han, J., Zhou, P., Zhang, D., Cheng, G., Liu, Z., Bu, S., Wu, J.: Efficient, simultaneous detection of multi-class geospatial targets based on visual saliency modeling and discriminative learning of sparse coding. ISPRS J. Photogramm. Remote Sens. **89**, 37–48 (2014)
15. Verma, N.K, Sunny, N.K, Mishra, A.: Generation of future image frame using autoregressive model. In: IEEE Conference on Industrial Electronics and Applications, pp. 171–176. Auckland, New Zealand (2015)

16. Verma, N.K, Mishra, A.: Large displacement optical flow based image predictor model. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, Oct 2014
17. Verma, N.K, Singh, S.: Generation of future image frames using optical flow. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, 23–25 Oct 2013
18. Kim, T.K., Paik, J.K., Kang, B.S.: Contrast enhancement system using spatially adaptive histogram equalization with temporal filtering. IEEE Trans. Consum. Electron. **44**(1), 82–87 (1998)
19. Stark, J.A.: Adaptive image contrast enhancement using generalizations of histogram equalization. IEEE Trans. Image Process. **9**(5), 889–896 (2000)
20. Kim, J.-Y., Kim, L.-S., Hwang, S.-H.: An advanced contrast enhancement using partially overlapped sub-block histogram equalization. IEEE Trans. Circuits Syst. Video Technol. **11**(4), 475–484 (2001)
21. Schechner, Y.Y., Narasimhan, S.G., Nayar, S.K.: Instant dehazing of images using polarization. In: Proceedings IEEE Conference Computer Vision Pattern Recognition (CVPR), pp. I-325I-332 (2001)
22. Shwartz, S., Namer, E., Schechner, Y.Y.: Blind haze separation. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 1984–1991 (2006)
23. Schechner, Y.Y., Narasimhan, S.G., Nayar, S.K.: Polarization-based vision through haze. Appl. Opt. **42**(3), 511525 (2003)
24. Nayar, S.K., Fang, X.S., Boult, T.: Separation of reflection components using color and polarization. Int. J. Comp. Vis. **21**, 163–186 (1997)
25. Saito, M., Sato, Y., Ikeuchi, K., Kashiwagi, H.: Measurement of surface orientations of transparent objects using polarization in highlight. Proc. CVPR **I**, 381–386 (1999)
26. Wolff, L.B.: Using polarization to separate reflection components. Proc. CVPR, 363–369 (1989)
27. Farid, H., Adelson, E.H.: Separating reflections and lighting using independent components analysis. Proc. CVPR, 262–267 (1999)
28. Schechner, Y.Y., Shamir, J., Kiryati, N.: Polarization based decorrelation of transparent layers: the inclination angle of an invisible surface. Proc. ICCV, 814–819 (1999)
29. Schechner, Y.Y., Shamir, J., Kiryati, N.: Polarization and statistical analysis of scenes containing a semi-reflector. JOSA-A **17**, 276–284 (2000)
30. Wolff, L.B.: Polarization vision: a new sensory approach to image understanding. Image Vis. Comput. **15**, 81–93 (1997)
31. Ben-Ezra, M.: Segmentation with invisible keying signal. Proc. CVPR, 32–37 (2000)
32. Shutov, A.M.: Videopolarimeters. Sov. J. Opt. Technol. **60**, 295–301 (1993)
33. Narasimhan, S.G., Nayar, S.K.: Chromatic framework for vision in bad weather. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 598–605 (2000)
34. Nayar, S.K., Narasimhan, S.G.: Vision in bad weather. In: Proceedings IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 820–827 (1999)
35. Narasimhan, S.G., Nayar, S.K.: Contrast restoration of weather degraded images. IEEE Trans. Pattern Anal. Mach. Intell. **25**(6), 713–724 (2003)
36. Narasimhan, S.G., Nayar, S.K.: Interactive (de) weathering of an image using physical models. In: Proceedings IEEE Workshop Color Photometric Methods Computer Vision, vol. 6, p. 1. France (2003)
37. Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D.: Deep photo: model-based photograph enhancement and viewing. ACM Trans. Graph. **27**(5), 116 (2008)
38. Agrawal, P., Sharma, T., Verma, N.K.: Supervised approach for object identification using speeded up robust features. Int. J. Adv. Intell. Paradig. (IJAIP) (Accepted for publication)
39. Verma, N.K., Sharma, T., Sevakula, R.K., Salour, A.: Vision based object counting using speeded up robust features for inventory control. In: International Conference on Computational Science and Computational Intelligence (CSCI '16). Las Vegas, Nevada, USA, 15–17 Dec (2016) (In Proceedings)

40. Verma, N.K., Sharma, T., Rajurkar, S.D., Salour, A.: Object identification for inventory management using convolutional neural network. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR). Washington DC, USA, 18–20 Oct (2016) (In Proceedings)
41. Verma, N.K., Sharma, T., Rajurkar, S.D., Ranjan, R., Salour, A.: Vision based counting of texture-less objects using shape and color features. In: IEEE International Conference on Industrial and Information Systems (ICIIS). IIT Roorkee, India, 3–4 Dec (2016) (In Proceedings)
42. Verma, N.K., Sharma, T., Rajurkar, S.D., Molangur, N., Salour, A.: Multi-faced object recognition in an image for inventory counting. In: IEEE International Conference on Design and Management (IConDM). IIITDM Kancheepuram, Chennai, India, 16–17 Dec (2016) (In Proceedings)
43. Verma, N.K., Goyal, A., Vardhan, A.H., Sevakula, R.K., Salour, A.: Object matching using speeded up robust features. In: 19th INNS Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES15), pp. 415–427. Bangkok, Thailand (2015)
44. Vardhan, A.H., Verma, N.K., Sevakula, R.K., Salour, A.: Unsupervised approach for object matching using speeded up robust features. In: IEEE Applied Imagery Pattern Recognition Workshop, (AIPR), pp. 1–8. Washington DC, USA (2015)
45. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. In: Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition (2009)
46. Goldstein, E.B.: Sensation and Perception. Wadsworth Publishing Co., Inc, Massachusetts (1980)
47. Xiang, W., Xiao, J., Wang, C., Liu, Y.: A new model for daytime visibility index estimation fused average sobel gradient and dark channel ratio. In: International Conference on Computer Science and Network Technology (2013)
48. Akilov, G.P., Kantorovich, L.V.: Functional Analysis, Pergamon Pr; 2 Sub edition (1982). ISBN 0-08-023036-9
49. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6) (2013)

Multi-faced Object Recognition in an Image for Inventory Counting

Teena Sharma, Shreedharkumar D. Rajurkar, Nikhil Molangur, Nishchal K. Verma and Al Salour

Abstract Herein, an approach is presented to count objects in an image with different viewpoints. The multiple views of prototype have been used to get the viewpoint invariance in Speeded Up Robust Feature. For each view, presence of multiple instances in the scene image is investigated by matching the SURF features. If matches present, localization of instance is done in scene by generating a bounding box using homography. False matches are removed by computing correlation coefficient between transformed prototype and region of interest in scene. Different views of same prototype leads to multiple bounding boxes representing same object instance in scene. A supervised learning approach is used for classification of bounding boxes representing same instance. Hence, bounding boxes are grouped. Finally, a single bounding box which best describes an instance is chosen. The proposed algorithm is able to count objects for different viewpoints with better accuracy in multiple cases.

Keywords Object counting · SURF · DBSCAN · Artificial neural network Perspective invariant features

T. Sharma (✉) · S. D. Rajurkar · N. K. Verma
Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: tee.shar6@gmail.com

S. D. Rajurkar
e-mail: shreedr@iitk.ac.in

N. K. Verma
e-mail: nishchal.iitk@gmail.com

N. Molangur
Indian Institute of Technology Bhubaneswar, Bhubaneswar 751013, India
e-mail: nm15@iitbbs.ac.in

A. Salour
The BOEING Company, St. Louis, MO, USA
e-mail: al.salour@boeing.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_26

1 Introduction

Object counting involves estimation of the number of objects of interest in an image. It has quite a lot of applications which include cell counting [1], transport management system [2], future image frame prediction [3–5], object counting for inventory management [6–10] and surveillance [11]. An automated approach to maintain record of the stock in inventory helps in efficient functioning of the inventory. It enables proper supply of goods based on the demand improving productivity [12]. A computer vision-based approach is taken to count the objects in the inventory and to localize their instances in the image.

Various approaches have been taken to count objects of interest in the image using shape information [13], background subtraction in vehicle detection and counting systems [14]. Image descriptors such as SIFT [15], SURF [16] are also used for object recognition purposes. In this paper, SURF has been used as feature descriptor as SURF features provide rotation, scale invariance and fast computing. Our previous works were based on colour histogram [17], fuzzy colour histogram [17], SURF-based recognition using support vector machine for classification [18]. The previous work [19] using SURF and DBSCAN clustering [20] failed to take into account the viewpoint variance of SURF even though it achieved scale, rotation, and illumination invariance. It is important to overcome this problem so as to detect object from any view. Also, it failed to achieve multi-faced object recognition even after considering multiple faces of the object as prototype samples. It could not discriminate appropriately between intra-class features and inter-class features. It is important to overcome this problem so as to detect the object from any of its faces.

In proposed approach, various viewpoints have been considered to resolve SURF's limitation of viewpoint variance. Various perspective transformations [21] of prototype image are matched with the scene image. Clustering techniques [20] are used to obtain different instances of a view of the prototype (in the form of bounding boxes). The obtained bounding boxes have to be grouped to reflect the object instance they represent. This is a difficult and challenging task. Two bounding boxes obtained from two views may represent the same face of an object instance. Also, two bounding boxes may represent two different faces of the same object. Grouping them into instances is carried out with the aid of artificial neural network (ANN) [22, 23] classifiers. The ANNs capture the intra-class as well as inter-class difference separately and help to achieve appropriate classification. Finally, the number of groups so formed gives the count of object instances in the scene.

The rest of the paper is organized as follows. Section 2 highlights the proposed approach to count object in inventory irrespective of viewpoint. Section 3 contains the information relating the experiments performed and results obtained. Section 4 concludes the work.

2 Methodology

Multi-face object detection comprises of object counting using SURF features irrespective of the view and face of prototype. In this section, the steps for object identification and counting using SURF and artificial neural network have been explained.

2.1 Prototype Modelling

Multi-face object detection involves use of faces of the object which have sufficient number of SURF features. The orientations of the object introduce a difficulty in the form of SURF's variance to perspective.

In affine-invariant SIFT (ASIFT) [24], views were simulated on the basis that local deformations due to viewpoint changes can be approximated by affine transformations as long as the objects are locally smooth. We have used views which relate to the frontal view of a prototype image Fig. 1 by a projective transform.

The views simulated are as follows:

1. $\theta = 0°$ and $\phi = 0°$
2. $\theta = 60°$ and $\phi = \dfrac{n\pi}{4}$ radians $\forall n \in [0, 7]$

θ is the angle made by camera axis with the normal to the image plane and ϕ is the angle made by projection of camera axis on image plane with some reference line on the image plane. So there are nine views of a face used for matching purposes.

To determine multiple faces and views of the prototype in scene image, nine views of each face of the prototype are generated based on transformation matrices. We have used SURF features to characterize an image primarily because of the speed at which these features can be computed and are good enough for scale and rotation variances. The SURF features and the coordinates of the corners of the objects in these views in images are computed. So a prototype is described by its six faces with each face being described by nine views. Six views of a face of prototype are shown in Fig. 2.

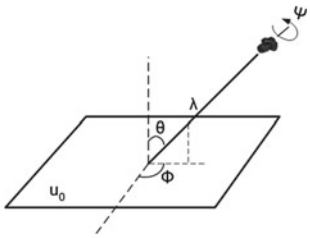


Fig. 1 Front view of flat object [24]

Fig. 2 Prototype views

SURF points are detected in scene image and features are extracted at detected points. The descriptors are found, and matching is done. It is observed some of the points are stable and appear at matching between various scales.

Algorithm 1: Modelling prototype

Load F_I; %Face Images
for i=1:num_faces %number of faces specified (usually 6)
  for j=1:num_views
    %number of views of a single face of object
      img = generateimagewithtransform($F_I(i)$, T(j));
      %img is j^{th} view of ith face
      protos(i).append(img);
      coords(i).append(extractCoordinates($F_I(i)$, T(j)))
      surfPoints(i).append(detectandextractSURF(img));
  end
end
save protos, coords, surfPoints

2.2 Inliers Between Scene Image and Prototype Model

Compared to the previous method [19], this involves more computation time due to the multiple views of a face. All the faces may not contribute to detection of an object in scene. These views need not be processed further. RANSAC [25, 26] is capable of detecting a model even in presence of large number of outliers. This property of RANSAC [26] (or its variant used (MLESAC) [27] is exploited to reduce computation time by filtering out faces which may not contribute to object in the image. Matching between prototype and scene features was done. This gives the number of inliers. The threshold for inliers filtering was kept between 4 and 7 (minimum number of points to determine homography is 4). For objects with rich texture, higher threshold improves the performance of the system. So the views which do not meet the threshold requirement are discarded.

2.3 Matching Between Prototype Model and Scene Image

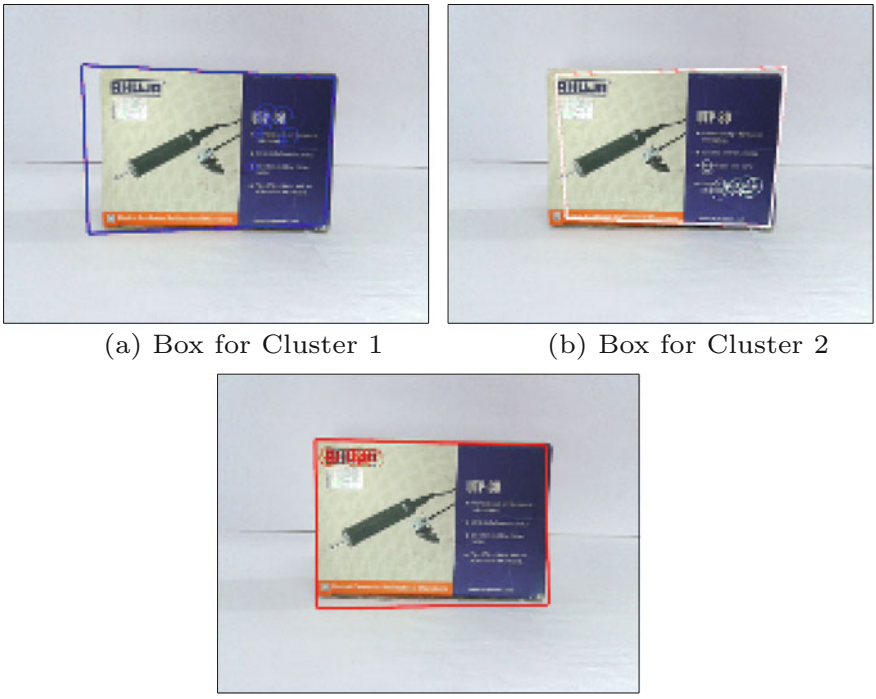
Matching between the scene image and the views of prototype (selected from above) is done. The matching factor is defined in terms of Sum of Squared Difference (SSD) and nearest neighbourhood ratio (NNR). NNR is the ratio of the SSD of a matching with SSD of next nearest matching. The SSD value less than a threshold of 1 is considered as matched pair, and the NNR of more than the threshold 0.6 is taken into consideration. The above values were suggested in SIFT [15] by Lowe.

2.4 DBSCAN Clustering

The SURF point locations of prototype are clustered by DBSCAN algorithm [20]. The matched SURF points in the scene for each cluster of prototype are again clustered using DBSCAN. Note this is a many-to-one mapping. Each cluster in scene is some instance of the object. The parameter K (minimum number of neighbours for clustering) is chosen to be 3.

2.5 Bounding Box Formation for Clusters in Scene Image

For each cluster in the view of prototype and corresponding clusters in scene, a homography (transformation matrix) is estimated. Now the corners of the prototype image mentioned earlier are projected by the computed matrix to form bounding box. The box polygon formed for each cluster in the scene is expected to contain an



(a) Box for Cluster 1 (b) Box for Cluster 2

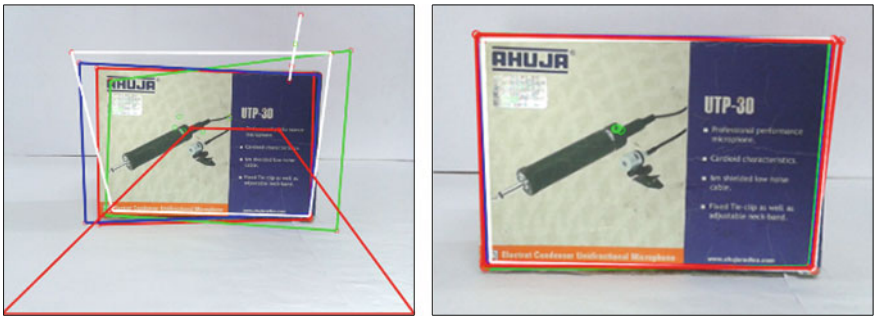
(c) Box for Cluster 3

Fig. 3 Different clusters and corresponding bounding boxes

instance of prototype. Figure 3 gives an example of how bounding boxes are formed corresponding to each cluster in the prototype.

2.6 Fixing of False Boxes

The inliers SURF points of box polygon formed in the scene are matched with the SURF points of view followed by estimating projective transformation matrix using matching pairs to form the true shape of box polygon. This step is repeated twice. If a box polygon is true, it will not be affected otherwise fixing is done by elimination or conversion of false box polygon into true box polygon. The fixing procedure of box polygon results in multiple boxes enclosing single object instance in scene due to various cluster obtained for a single object (Refer Fig. 4).



(a) Before fixing (b) After fixing

Fig. 4 Fixing of false boxes

(a) Bounding boxes (b) Final box

Fig. 5 Merging of boxes

2.7 *Merging of Boxes*

For a single view, we have obtained multiple bounding boxes so the centroids of the obtained bounding boxes are clustered. Each cluster contains bounding boxes representing same object instance. Average of coordinates of corners of bounding boxes is used to represent the merged bounding box. Some of false boxes, such as concave bounding boxes, introduced during homography estimation may be eliminated based on geometry (Refer Fig. 5).

2.8 *Filtering Based on Correlation Coefficient*

In Fig. 2, we observe some repeated patterns in the object (such as alphabets). So it is expected to result in some false bounding boxes. This is taken care of by this step. The region defined by bounding box is extracted (Z_a), and the prototype view responsible

for the bounding box (Z_b) is transformed by computed projective transform. The correlation coefficient of these two images (Z_a) and (Z_b) is used to filter false boxes. Choosing a threshold involves trade-off. If threshold is set high, true matches with illumination changes are filtered out. If threshold is set low, false matches too are included. A threshold between 0.75 and 0.85 provides appreciable results. The correlation coefficient between (Z_a) and (Z_b) is defined as:

$$r = \frac{\sum_m \sum_n (Z_{amn} - \bar{Z}_a)(Z_{bmn} - \bar{Z}_b)}{\sqrt{(\sum_m \sum_n (Z_{amn} - \bar{Z}_a)^2)(\sum_m \sum_n (Z_{bmn} - \bar{Z}_b)^2)}} \tag{1}$$

2.9 Artificial Neural Network (ANN) Training

We now have bounding boxes corresponding to all views of faces of the object. Based on obtained boxes, we should be able to infer count. There may possibly exist any one of the following three relations between any pair of bounding boxes so obtained (Refer Fig. 6)

(a) Scenario (i) (b) Scenario (ii)

(c) Scenario (iii)

Fig. 6 Different situations during classification

Table 1 Labels given to different scenarios in Fig. 6

Scenario	Classifier 1	Classifier 2
(i)	False	True
(ii)	False	False
(iii)	True	False

1. The two boxes belong to different faces of an object instance (Scenario (i)).
2. The two boxes belong to different object instances in image (Scenario (ii)).
3. The two boxes belong to same face of an object instance in image (Scenario (iii)).

Thus, there may be multiple boxes describing the same object. Our task is to group bounding boxes representing same object instance together. This classification problem is solved by using two ANNs [22, 23]. One decides if the bounding boxes belong to the same face of an object, i.e. capturing process and processing intra-class difference. Other decides if the bounding boxes belong to different faces of same object i.e. capturing the inter-class (two different faces) difference. The decisions above are based on orientation of bounding boxes under observation. If both the decisions are false, it implies that the bounding boxes under consideration belong to different object instances in the scene image.

Figure 6 and Table 1 give an idea of how labels are marked for different scenarios. Classifier 1 represents the first artificial neural network layer and Classifier 2 represents the second artificial neural network layer.

Algorithm 2: Grouping of Bounding boxes using ANN

```
Consider an array of size numBB
for i=1:numBB %Number of bounding boxes
  for j=i+1:numBB
    a=BB(1); b=BB(2);
    featureVec=constructFeature(a, b);
    flag=(network(featureVec)>=th);
    if flag
      if consider(j)==0
        consider(i)=0; class(i)=class(j);
        continue
      end
      consider(j)=0; class(j)=class(i);
    end
  end
end
Object's first instance as 1 and remaining 0;
count = sum(instance as 1);
```

Algorithm 3: Object Identification and Counting

```
Step 1: Prototype model and scene's SURF features
Step 2: Estimate number of inliers and pass those images with higher no. of inliers
Step 3: for i=1:numFaces
          for j=1:numViews %j view of i face
            view = prototype{i}{j};
            P_s, S_s : SURF points of proto and scene
            G_p : Clusters formed in proto
            G_s : Groups of matched SURF points in scene
           // Each group corresponds to match with one cluster of proto
            Clustering each G_s
            Fixing and merging of boxes drawn in scene
            boundBox.append (filtered bounding boxes);
          end
        end
Step 4: for i=1:length(boundBox)
          box=boundBox(i);
          if(r(proto, roi(scene, box)) < th)
             discard box;
          end
        end
Step 5: class1 = CombineResults(boundBox, network1)
        class2 = CombineResults(boundBox,network2)
Step 6: class = connected(class1,class2)
        %gives the final estimated class
Step 7: No. of classes = No.of object instances
```

3 Results

In Fig. 6, we observe a single face being represented by multiple boxes and each object detected at multiple faces. This is because of the different views of the face contributing a bounding box. Based on training data, bounding box is selected from each group to represent the corresponding object instance in scene image. The results are shown in Fig. 7.

Table 2 compares the performance of the proposed approach with previous one [19]. The increase in accuracy can be easily depicted by observing the decrease in miss ratio. For better performance of proposed algorithm, miss ratio should be less. The miss ratio is defined as follows:

(a) Sample 1 (Before) (b) Sample 1 (After)

(c) Sample 2 (Before) (d) Sample 2 (After)

(e) Sample 3 (Before) (f) Sample 3 (After)

(g) Sample 4 (Before) (h) Sample 4 (After)

(i) Sample 5 (Before) (j) Sample 5 (After)

Fig. 7 Results for various test cases using SURF clustering method [19] and proposed approach

Table 2 Count of objects in Fig. 7

Test cases	Actual count	SURF clustering method		Proposed approach	
		Estimated count	Miss ratio	Estimated count	Miss ratio
Sample 1	1	1	0	1	0
Sample 2	3	0	1	2	0.33
Sample 3	3	1	0.67	3	0
Sample 4	3	2	0.33	3	0
Sample 5	3	1	0.67	3	0

$$\text{Miss Ratio} = \frac{\text{Actual Count} - \text{Estimated Count}}{\text{Actual Count}} \tag{2}$$

Figure 7 depicts the results for previous approach [19] and proposed approach. From obtained results, we can see that we are able to detect the bounding boxes fairly well and also group them into the object instances they represent.

4 Conclusions

The proposed approach is able to detect the count of the instances of prototype in the scene image obtained from inventory irrespective of the viewpoint variance in prototype. It performs reasonably well with rotation and scale invariance. It also gives the approximate bounding box which closely localizes the object instance in inventory. The supervised learning approach using artificial neural network provides promising results to distinguish the true and false alarms. The artificial neural network may further be benefited by including the label of face represented by bounding box.

Acknowledgements Authors would like to express their sincere thanks towards The BOEING Company, USA, for sponsoring this research work at IIT Kanpur.

References

1. Danuser, G.: Computer vision in cell biology. Cell **147**(5), 973–978 (2011)
2. Coifman, B., Beymer, D., McLauchlan, P., Malik, J.: A real-time computer vision system for vehicle tracking and traffic surveillance. Transp. Res. Part C: Emer. Technol. **6**(4), 271–288 (1998)
3. Verma, N.K., Sunny, N.K., Mishra, A.: Generation of future image frame using autoregressive model. In: IEEE Conference on Industrial Electronics and Applications, pp. 171–176. Auckland, New Zealand, (2015)

4. Verma, N.K., Mishra, A.: Large displacement optical flow based image predictor model. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, Oct 2014
5. Verma, N.K., Singh, S.: Generation of future image frames using optical flow. In: Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, 23–25 Oct 2013
6. Agrawal, P., Sharma, T., Verma, N.K.: Supervised approach for object identification using speeded up robust features. Int. J. Adv. Intell. Paradigims (IJAIP) (2017) (Accepted for publication)
7. Verma, N.K., Sharma, T., Sevakula, R.K., Salour, A.: Vision based object counting using speeded up robust features for inventory control. In: International Conference on Computational Science and Computational Intelligence (CSCI'16). Las Vegas, Nevada, USA, 15–17 Dec (2016) (In Proceedings)
8. Verma, N.K., Sharma, T., Rajurkar, S.D., Salour, A.: Object identification for inventory management using convolutional neural network. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR). Washington DC, USA, 18–20 Oct (2016) (In Proceedings)
9. Verma, N.K., Sharma, T., Rajurkar, S.D., Ranjan, R., Salour, A.: Vision based counting of texture-less objects using shape and color features. In: IEEE International Conference on Industrial and Information Systems (ICIIS). IIT Roorkee, India, 3–4 Dec (2016) (In Proceedings)
10. Verma, N.K., Sharma, T., Rajurkar, S.D., Molangur, N., Salour, A.: Multi-faced object recognition in an image for inventory counting. In: IEEE International Conference on Design and Management (IConDM). IIITDM Kancheepuram, Chennai, India, 16–17 Dec (2016) (In Proceedings)
11. Lempitsky, V., Zisserman, A.: Learning to count objects in images. Adv. Neural Inf. Process. Syst. 1324–1332 (2010)
12. Zipkin, P.H.: Foundations of Inventory Management. Mc-Graw-Hill, New York (2000)
13. Whlby, C., Sintorn, I.M., Erlandsson, F., Borgefors, G., Bengtsson, E.: Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. J. Micros. **215**(1), 67–76 (2004)
14. Mandellos, N.A., Keramitsoglou, I., Kiranoudis, C.T.: A background subtraction algorithm for detecting and tracking vehicles. Exp. Syst. Appl. **38**(3), 1619–1631 (2011)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
16. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: speeded up robust features. Comput. Vis. ECCV, 404–417 (2006)
17. Verma, N.K., Goyal, A., Chaman, A., Sevakula, R.K.: Template matching for inventory management using fuzzy color histogram and spatial filters. In: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 317–322 (2015)
18. Verma, N.K., Goyal, A., Vardhan, A.H., Sevakula, R.K., Salour A.: Object matching using speeded up robust features. Intell. Evolut. Syst. 415–427 (2016)
19. Vardhan, A.H., Verma, N.K., Sevakula, R.K., Salour, A.: Unsupervised approach for object matching using speeded up robust features. In: 2015 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–8 (2015)
20. Ester, M., Kriegel, H.P., Sander, J., Xiaowei, X.: A density based algorithm for discovering clusters in large spatial databases with noise. KDD **96**(34), 226–231 (1996)
21. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge university press (2003)
22. Haykin, S.: Neural networks. A comprehensive foundation. Neural Netw. **2** (2004)
23. Kline, D.M., Victor, L.B.: Revisiting squared-error and cross-entropy functions for training neural network classifiers. Neural Comput. Appl. **14**(4), 310–318 (2005)
24. Morel, J.M., Guoshen, Y.: ASIFT: a new framework for fully affine invariant image comparison. SIAM J. Imag. Sci. **2**(2), 438–469 (2009)
25. Fischler, M.A., Robert, C.B.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)

26. Schnabel, R., Wahl, R., Klein, R.: Efficient RANSAC for point cloud shape detection. Comput. Graph. Forum **26**(2), 214–226 (2007)
27. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Understand. **78**(1), 138–156 (2000)

Detection of Dust Deposition Using Convolutional Neural Network for Heritage Images

Teena Sharma, Pooja Agrawal and Nishchal K. Verma

Abstract This paper presents a vision-based approach for heritage image classification and condition monitoring to preserve the historical facts. The proposed approach uses convolutional neural network for classification. The approach interprets the heritage condition in terms of dust level. Initially, real-time scene image is preprocessed using image processing operators such as dilation, erosion, region filling, and binarization. Resultant image is segmented and enclosed by bounding boxes. The enclosed segments are fed to CNN for classification. The proposed approach also provides the dust level in image by comparison of probability score of the classified image with ideal one. The dust is interpreted as Gaussian noise in the image. The dust level, greater than an acceptable tolerance level, generates a notification for heritage maintenance. Results show that the proposed approach is able to classify the heritage image in the presence of noise.

Keywords Heritage maintenance · Preprocessing · Segmentation Classification · CNN · Softmax · Gaussian noise · Dust identification

1 Introduction

Heritage maintenance is the process to protect the heritages as the physical presence of heritage, the root, and history of the society. Heritages represent the architectural, historical, cultural, spiritual, political, social, and archaeological values [1]. It can be characterized by corrective, emergency, and planned maintenance. Continuous

T. Sharma (✉) · P. Agrawal · N. K. Verma
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: tee.shar6@gmail.com

P. Agrawal
e-mail: pooja.iitd@gmail.com

N. K. Verma
e-mail: nishchal.iitk@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_27

monitoring of heritages can preserve the environmental conditions such as sunlight, wind, rain. These conditions can deteriorate and reduce the life of heritages. Regular and routine maintenance of heritage reduces the costly repairs and protection. The maintenance also includes non-destructive actions, cyclic inspection, cleaning, repair, finishing of historic materials, and monuments.

Heritage provides us the information about work, technology, and social history of our ancestors. The maintenance of heritage is challenging worldwide. The damage and deterioration in heritages can be characterized as dust, crack, dullness, etc. The laser Dopper vibrometer [2] is able to identify the zones which are suffering from damage in the historical buildings due to structural decay. Laser Dopper vibrometer estimates the accurate and reliable displacement value in these zones. In the literature, some researchers had used the wireless sensors for identifying the damage zones, where it is unable to find out the cracks manually [3]. In [4], the monitoring of historical buildings is explored using conventional and optical fiber sensors. Cole et al. have proposed a method for corrosion prediction on metallic materials in [5], which can be utilized in maintenance of heritages for identification of deterioration and damage of materials due to external factors.

Automation in heritage maintenance can be easily achieved by various techniques such as different sensors can be placed at different locations for identifying damaged zones. Automated maintenance is not only limited to these techniques, it can be extended using computer vision techniques. Computer vision act as human eye and analyzes the real-time images according to the application field. Image can be captured using vision interface (wired webcam, IP camera, etc.). Nowadays, computer vision-based approaches are being used in various sectors such as food and agriculture [6], traffic monitoring [7], textiles [8], surveillance [9], counting [10], future image frames prediction [11–13]. Various image processing techniques can be applied for identification of required facts as human eye characterizes the real-time scene using shape [14, 15], color [16], texture [17], etc. These image processing techniques include edge detection [18], image segmentation [19], image classification [20], etc.

Herein, a computer vision-based approach has been used to identify the dust deposition in heritage images. The real-time scene is captured using vision interface. Image preprocessing operations are applied on the input images to remove the noise and to enhancement. The preprocessed image is segmented to extract out the connected regions or to discriminate the different materials presented in the image. Each segmented regions are fed to convolutional neural network (CNN) for feature extraction followed by softmax classifier for calculation of probability score. It is compared with the trained model of labeled dataset. Finally, the dust level is identified by comparison of two probability scores: (1) probability score obtained in the presence of noise and (2) probability score obtained in the absence of noise. The Gaussian filter is also introduced for removal of noise incurred due to vision interface.

Further, the paper consists of subsections as follows: Sect. 2 presents image preprocessing to enhance the local features of image. Section 3 discusses image segmentation to extract out the connected regions based on some attributes in the image.

Section 4 gives image classification to identify the segmented region using convolutional neural network (CNN) followed by softmax classifier. The datasets used for CNN training and softmax classifier are also shown. Noise-level identification to calculate the dust level in heritages is provided in Sect. 5. Section 6 consists some results obtained during testing phase. Finally, Sect. 7 concludes the work done with its limitations and future challenges.

2 Image Preprocessing

2.1 Dilation

Dilation [21] is a mathematical operation in image processing for expansion of boundary regions. Consider, E-Euclidean coordinates set, k-Kernel coordinates set, ke-translation of k with origin e, then dilation of E by k is set of all points e with non-empty value of intersection of Ke with E. Dilation of E by k can be written as follows:

$$E \oplus k = \bigcup_{k' \in k} E_{k'} \tag{1}$$

where, $E_{k'}$ is E translated by k'. Figure 1a–c show the input RGB image, grayscale image, and dilated image, respectively.

(a) Input RGB image (b) Grayscale image (c) Dilated image

Fig. 1 Dilation on single-instance image

(a) Erosion on single instance image (b) Subtracted image (c) Binarized image

Fig. 2 Erosion and binarization on grayscale image

2.2 Erosion

Erosion [21] is the replica of dilation and can also be applied on binary and grayscale images. The basic effect of erosion in an image is that it shrinks the size of the image boundary regions. Erosion of E by k kernel can be defined as follows:

$$E \ominus k = \{z \in Z | k_z \subseteq E\} \tag{2}$$

where k_z is k translated by vector z. This translation can be written as follows:

$$k_z = \{k' + z | k' \in k\} \forall z \in Z \tag{3}$$

Figure 2a shows the erosion operation on a grayscale image. Figure 2b subtraction of eroded image from dilated image and binary image is shown in Fig. 2c.

2.3 Region Filling

Region filling [22] operates on background pixels of binary image and helps to reduce the small regions which act as noise points in the image by searching for boundaries in the image with closed contours. If a boundary in an image for region R is denoted by $B(R)$ and initial location is $y = y_o$, then filled region can be calculated as follows:

$$Y_k = (Y_{k-1} \oplus B) \cap R^C, \qquad k = 1, 2, 3, \ldots \tag{4}$$

This will continue until $Y_k = Y_{k-1}$. Here, R^C represents the complement of region R. The filled region is as follows:

$$R \cup Y_k \tag{5}$$

Figure 3a shows the image generated after erosion filling operation.

3 Image Segmentation

The objective of the image segmentation is to find out the appropriate segments in an image with similar attributes. These attributes are color [16], shape [14, 15], texture [17], etc. The grouped regions have non-intersecting and homogeneous nature. In [23], various image segmentation techniques are introduced with their performance.

3.1 Connected Component Analysis (CCA)

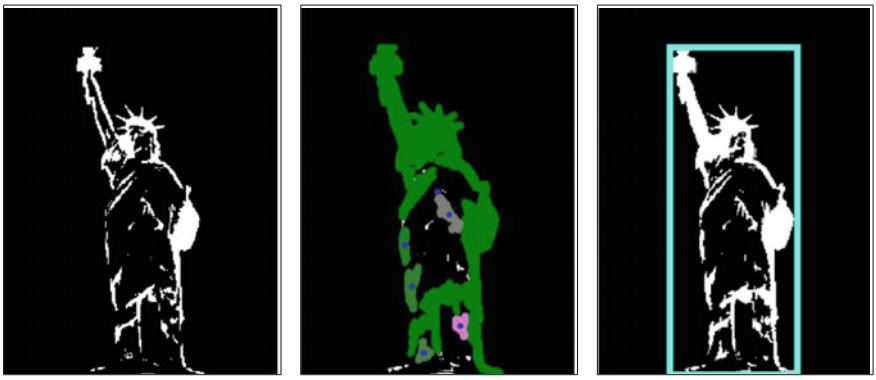
Connected component analysis (CCA) is the labeling process [24], which extracts the connected regions from the binary image. It scans each pixel of input image until a location p is found. Here, p denotes the pixel location to be labeled for which intensity value is 1 (since, binary image contains only 0 (black) or 1 (white) intensity values). When scanning reaches to the pixel p, the approach compares the pixel with the neighborhood pixels, i.e., above (a_p), left l_p and left two diagonal pixels (ld_{p1}, ld_{p2}). This is followed by labeling such as:

- if $a_p = l_p = ld_{p1} = ld_{p2} = 0$
  Assign p as new label.
- if a_p or l_p or ld_{p1} or ld_{p2} = 1 (i.e., any one from these)
  Assign p as label.
- if more than one is having value 1
  Assign p as label to one and write others in equivalences data.

The connected regions obtained after connected component analysis (CCA) are shown in Fig. 3b.

3.2 Bounding Box Formation

The connected regions obtained after connected component analysis (CCA) are further processed to find the bounding box around them. Bounding box formation is required to extract out the segments obtained from connected regions. Initially, bounding box approach removes the connected regions having very less pixels as



(a) Region filling (b) CCA after region filling (c) Bounding box formation

Fig. 3 Region filling, CCA and bounding box formation on an image

this may occur due to noise in the image. Next, the small square bounding boxes of dimensions 40×40 or 80×80 are drawn around the center of connected regions. The overlap area among these bounding boxes represents that whether two boxes are forming same region or different. This overlap in small bounding box area is a heuristic parameter which can be varied based on the resolution and information content of the image. Hence, final bounding boxes can be drawn for the connected regions. Figure 3c shows the bounding box drawn around the connected regions obtained using CCA.

4 Image Classification

The process of assigning a label to the input image from the set of labels is known as image classification. It can be done on the basis of visual content such as edges, color, texture, shapes [25, 26]. Nowadays, handcrafted features are in practice for real-time identification. These are scale invariant feature transform (SIFT) [27], speeded up robust feature (SURF) [28], etc. In [29, 30], real-time experimentation using SURF has been done. As number of images increases, these features become computationally expensive and unable to perform well. This limitation can be solved by learning the image features by deep neural network. Deep neural networks deal with features extracted from multiple images using trained layer and corresponding weights. Although deep neural networks perform well, it is computationally intensive during training phase for high-resolution images. This problem can be dealt with convolutional neural network [31].

Fig. 4 Dataset of historical monuments of seven different classes

4.1 Training Datasets

Herein, image classification has been done on the heritages. Figure 4 shows the dataset prepared for the images of historical monuments. It has seven classes in it, and each class is having 20 images for different view angles and illuminations for training. In Fig. 4, ten images for each class have been shown.

4.2 Convolutional Neural Network (CNN)

Neural network is the connected network having structure such that each layer is connected with the other layer. The neural network can be elaborated as, how each neuron is connected with other neurons defined by some weight. A simple neural network [32] is feedforward neural network. It has an input layer, output layer, and hidden layer. The connection starts from input nodes and ends at output nodes. The output generated from output layer is fed back for weight adjustment using back-propagation. The weight adjustment is done such that error between the input and output must be as minimum as possible. This process is continued until weights are not being adjusted. Figure 5 shows the architecture of convolutional neural network

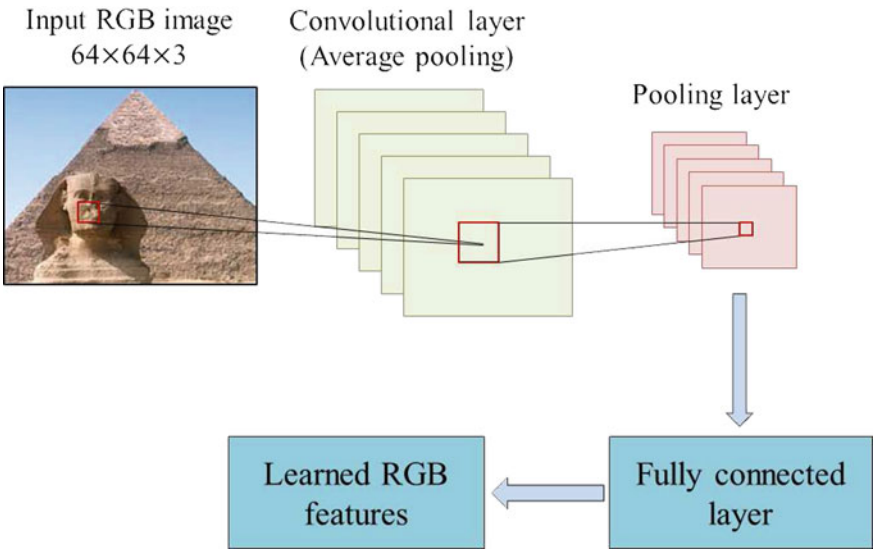


Fig. 5 Architecture of convolutional neural network (CNN)

(CNN). It consists of an input image, convolutional layer, sub-sampling or pooling layer, and fully connected layer.

CNN architecture is made up of various convolutional and sub-sampling layer. Herein, single-layer architecture, i.e., one convolutional layer and one sub-sampling layer followed by fully connected layer, is used for image classification. The input image of CNN is of size $s \times s \times c$, where c denotes the number of channels of input image. The filter kernel for convolutional layer is of size $p \times p \times h$, where p must be less than s, i.e., image dimension. The features' maps generated after convolutional layer is of size $s - p + 1$, and number of total feature maps are f. These feature maps are pooled for reducing the features. The pooling can be max, min, or average pooling. After this, fully connected layer is present.

Back-Propagation Algorithm It is the process of minimizing the difference between the values of input layer and output layer. It can be termed as cost function. If $\delta^{(t+1)}$ is error with cost function $J(W, b; d, l)$ of layer $t + 1$, where, W is weight parameter, b is bias, d and l are training data and its label, respectively. If there exists a dense connection between layer t and $t + 1$, then error for layer t is as follows:

$$\delta^t = \left(\left(W^t\right)^T \delta^{t+1}\right) \cdot I'\left(m^t\right) \tag{6}$$

The gradients are as follows:

$$\Delta_{W^{(t)}} J\left(W, b; d, l\right) = \delta^{(t+1)} \left(a^{(t)}\right)^T \tag{7}$$

$$\Delta_{b^t} J\left(W, b; d, l\right) = \delta^{t+1} \tag{8}$$

The error is propagated as follows:

$$\delta_i^{(t)} = upsample\left(\left(W_i^{(t)}\right)^T \delta_i^{(t+1)}\right) \cdot I'\left(m_i^{(t)}\right) \tag{9}$$

where i is filter number indices and $I'\left(m_i^{(t)}\right)$ is derivative of activation function.

4.3 Softmax Classifier

Classifier is basically used for classification purpose based on some trained model. Various classifiers are now being used in identification such as support vector machines (SVMs) [33], softmax [34]. During training step of linear sparse autoencoder, use of labeled dataset makes classification expensive because trained model has to remember the labels for each annotations. In softmax classifier, multi-class classification can be done with respect to binary classification. It focuses on two major components mainly: (1) classification score (2) loss function. The classification score calculated for input image is compared with the labeled values for which loss function is minimized. The loss function is defined in order to estimate the value of parameters with respect to ground truth value for the training models.

5 Noise-Level Identification

Noise can be interpreted as random signals which are not actually true signal values. It is also known as unwanted signals. In image processing, noise is the variation in brightness and color intensity values which are not part of actual intensity values of an object or a material. Herein, an image processing is being utilized for heritage maintenance. Dust in the heritage images can be modeled as noise. If an image segment I_s with crack and dust as noise has probability score P_N and same image segment I_s without noise has probability score P, then the ratio of these two is as follows:

$$\eta = \frac{P - P_N}{P} \times 100\% \tag{10}$$

The η represents the percentage of noise present in the input image with respect to the ideal image. It can also be characterized as the how much information is missed for classifying an image segment. Next section discusses the testing phase of the proposed approach.

6 Testing Phase

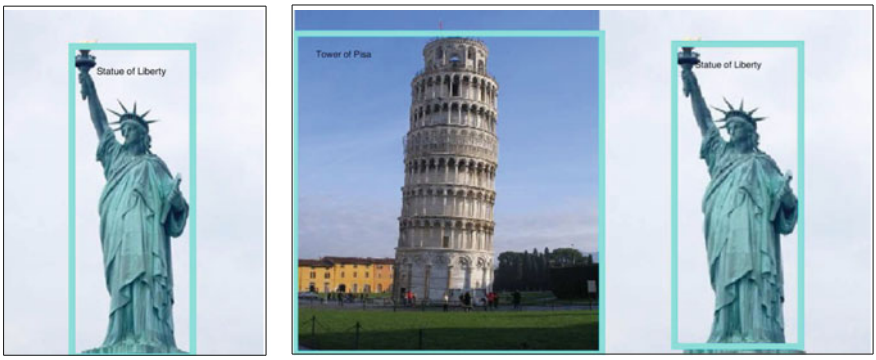
The image classification has been performed using dataset as shown in Fig. 4. The convolutional neural network (CNN) has been trained using STL-10 dataset [35]. The CNN is trained for input image of size $64 \times 64 \times 3$, where 3 denotes the number of channels in the input image. The size of hidden layer is 50 for generation of trained model. The weight matrix generated using STL-10 features is used to train softmax classifier. Softmax classifier is trained using the labeled dataset of 140 images of heritage. This dataset includes 7 different classes for training as shown in Fig. 4 (Fig. 6).

Gaussian Filter Gaussian Filter [36] performs smoothening in the image by reducing the gaussian noise. It can also be termed as blurring of the image. During this process, sometimes we may lost the information of fine edges or rapid intensity changes. The Gaussian distribution function in $2 - D$ is defined as follows:

$$G(x,\ y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{11}$$

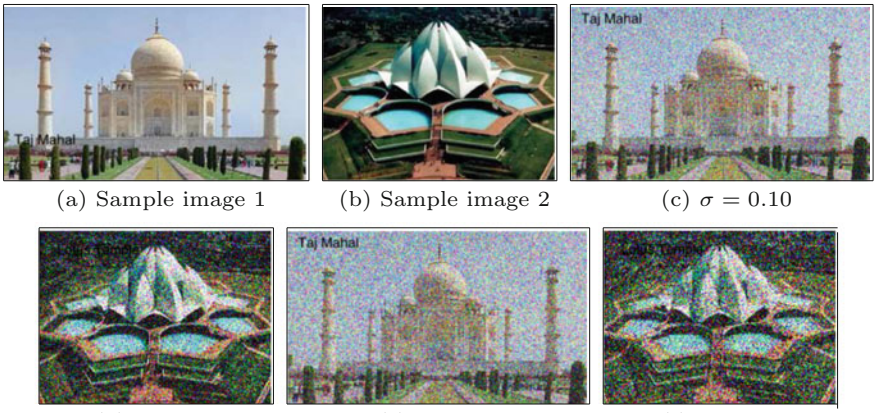
where σ is the standard deviation. In Eq. 11, the mean of the distribution is assumed as zero.

Figure 7a, b show the sample images of Taj Mahal and Lotus Temple, respectively. Noise level of these images is calculated using Eq. 10. The input image of Taj Mahal without noise (Fig. 7a) is corrupted with Gaussian noise with different values of σ is equal to 0.10 and 0.20 as shown in Fig. 7c, e, respectively. The percentage of noise levels for image of Taj Mahal and Lotus Temple with different noise levels is given in Table 1 and Table 2, respectively.



(a) Single instance classified image (b) Multiple instance classified image

Fig. 6 Results for single- and multiple-instance images



(a) Sample image 1 (b) Sample image 2 (c) $\sigma = 0.10$

(d) $\sigma = 0.10$ (e) $\sigma = 0.20$ (f) $\sigma = 0.20$

Fig. 7 Images with Gaussian noise for different values of σ

Table 1 Probability score comparison in the presence of Gaussian noise for class of Taj Mahal

Value of σ	Probability score (without noise)	Probability score (with noise)	Value of η (%)
0.10	0.9998462	0.9980992	0.14207
0.20	0.9998462	0.9961127	0.49523
0.30	0.9998462	0.9899812	0.98600
0.40	0.9998462	0.9844825	1.53662

Table 2 Probability score comparison in the presence of Gaussian noise for class of Lotus Temple

Value of σ	Probability score (without noise)	Probability score (with noise)	Value of η (%)
0.10	0.999882	0.9978582	0.009620
0.20	0.999882	0.9998398	0.009888
0.30	0.999882	0.9995672	0.031486
0.40	0.999882	0.9995369	0.034510

In Tables 1 and 2, the probability score obtained for input RGB image with gaussian noise, incurred for a value of σ, has been compared with ideal probability score. The noise level (η) in image of heritages can generate an alarm after an increase in dust level until an acceptable limit for dust level has been crossed.

7 Conclusions

In this paper, a robust heritage image classification and noise-level identification using convolutional neural network (CNN) are presented. The proposed approach utilizes the computer vision for heritage classification and maintenance. Results

show that the proposed approach was able to classify the heritage images in the presence of noise. Moreover, the approach is able to detect the amount of dust on historical materials, which can be incurred due to environmental conditions. The future work can be done for classification of cracks identification in heritage walls. For this purpose, the ideal image details have to be fed to softmax classifier along with labeled dataset.

References

1. Feilden, B., Jokilheto, J.: Management Guidelines for World Cultural Heritage Sites. ICCROM, Rome (1993)
2. Garziera, R., Amabili, M., Collini, L.: Structural health monitoring techniques for historical buildings. Eng. Struct. **19**(9), 718–723 (1997)
3. Anastasi, G., Re, G.L., Ortolani, M.: WSNs for structural health monitoring of historical buildings. In: 2nd IEEE Conference on Human System Interactions, pp. 574–579 (2009)
4. Glisic, B., Inaudi, D., Posenato, D., Figini, A., Casanova, N.: Monitoring of heritage structures and historical monuments using long-gage fiber optic interferometric sensorsan overview. In: Proceedings of the 3rd International Conference on Structural Health Monitoring of Intelligent Infrastructure-SHMII-3, pp. 13–16. Vancouver, BC, Canada (2007)
5. Cole, I.S., Corrigan, P.A., Ganther, W., Ho, T., Lewis, C.J., Muster, T.H., Galea, S.: Development of a sensor-based learning approach to prognostics in intelligent vehicle health monitoring. In: IEEE International Conference on Prognostics and Health Management, pp. 1–7 (2008)
6. Jafri, R., Arabnia, H.R.: A survey of face recognition techniques. JIPS **5**(2), 41–68 (2009)
7. Gonzlez, G.M., Llorca, D.F., Gaviln, M., Fernndez, J.P., Alcantarilla, P.F., De Toro, P.R.: Automatic traffic signs and panels inspection system using computer vision. IEEE Trans. Intell. Transp. Syst. **12**(2), 485–499 (2011)
8. Malek, A.S.: Online fabric inspection by image processing technology. Doctoral Dissertation, Universit de Haute Alsace-Mulhouse (2012)
9. Roy, S., Roy, S.: A tutorial review on face detection. Int. J. Eng. Res. Technol. (IJERT) **1**(8), 2278-0181 (2012)
10. Verma, N.K., Sharma, T., Rajurkar, S.D., Salour, A.: Object identification for inventory management using convolutional neural network. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR). Washington DC, USA (2016) (In Proceedings)
11. Verma, N.K., Sunny, N.K., Mishra, A.: Generation of future image frame using autoregressive model. In: IEEE Conference on Industrial Electronics and Applications, pp. 171–176. Auckland, New Zealand, (2015)
12. Verma, N.K., Mishra, A.: Large displacement optical flow based image predictor model. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, Oct 2014
13. Verma, N.K., Singh, S.: Generation of future image frames using optical flow. In: Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, 23–25 Oct 2013
14. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., HUang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: the QBIC system. IEEE. Comput. **28**, 23–32 (1995)
15. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 509–522 (2002)
16. Deng, Y., Manjunath, B.S., Kenney, C., Moore, M.S., Shin, H.: An efficient color representation for image retrieval. IEEE Trans. Image Process. **10**, 140–147 (2001)

17. Portilla, J., Simoncelli, E.: A parametric texture model based on joint statistics of complex wavelet coefficients. Int. J. Comput. Vis. **40**, 49–71 (2000)
18. Rosenfeld, A.: A nonlinear edge detection technique. Proc. IEEE **58**(5), 814–816 (1970)
19. Kuruvilla, J., Sukumaran, D., Sankar, A., Joy, S.P.: A review on image processing and image segmentation. In: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), pp. 198–203 (2016)
20. Moghaddam, A.A.: Image processing techniques for classification of linear welding defects. In: 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 978–981 (2015)
21. Chen, S., Haralick, R.M.: Recursive erosion, dilation, opening, and closing transforms. IEEE Trans. Image Process. **4**(3), 335–345 (1995)
22. Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9), 1200–1212 (2004)
23. Taneja, A., Ranjan, P., Ujjlayan, A.: A performance study of image segmentation techniques. In: 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), India (2015)
24. Kang, S.M., Kim, J.H., Yuan, Z., Song, S.H., Cho, J.D.: A fast region expansion labeling of connected components in binary image. In: 18th IEEE International Symposium on Consumer Electronics (ISCE) (2014)
25. Verma, N.K., Goyal, A., Chaman, A., Sevakula, R.K., Salour, A.: Template matching for inventory management using fuzzy color histogram and spatial filters. In: IEEE Conference on Industrial Electronics and Applications, pp. 317–322. AuckLand, New Zealand (2015)
26. Verma, N.K., Sharma, T., Rajurkar, S.D., Ranjan, R., Salour A.: Vision based counting of texture-less objects using shape and color features. In: IEEE International Conference on Industrial and Information Systems (ICIIS). IIT Roorkee, India (2016) (In Proceedings)
27. Lowe, D.G.: Object recognition from local scale-invariant features. In: 7th IEEE International Conference on Computer Vision, Kerkrya, Greece (1999)
28. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European Conference on Computer Vision, pp. 404–417. Springer, Berlin, Heidelberg (2006)
29. Verma, N.K., Sharma, T., Sevakula, R.K., Salour, A.: Vision based object counting using speeded up robust features for inventory control. In: International conference on Computational Science and Computational Intelligence (CSCI16), Las Vegas, Nevada, USA (2016) (In Proceedings)
30. Agrawal, P., Sharma, T., Verma, N.K.: Supervised approach for object identification using speeded up robust features. Int. J. Adv. Intell. Paradigms (IJAIP) (2016) (Accepted for publication)
31. Verma, N.K., Sharma, T., Rajurkar, S.D., Salour, A.: Object identification for inventory management using convolutional neural network. IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington DC, USA (2016) (In Proceedings)
32. Fukushima, K., Miyake, S.: Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and Cooperation in Neural Nets, pp. 267–285. Springer, Berlin, Heidelberg (1982)
33. Vapnik, V.N.: An overview of statistical learning theory. IEEE Trans. Neural Netw. **10**(5), 988–999 (1999)
34. Zhang, L., Huang, H., Jing, X.: A modified cyclostationary spectrum sensing based on softmax regression model. In: 16th International Symposium on Communications and Information Technologies (ISCIT), pp. 620–623 (2016)
35. Coates, A., Lee, H., Ng, A.Y.: An analysis of single layer networks in unsupervised feature learning. Ann. Arbor. **1001**(48109), 2 (2010)
36. Deng, G., Cahill, L.W.: An adaptive Gaussian filter for noise reduction and edge detection. In: IEEE Nuclear Science Symposium and Medical Imaging Conference, pp. 1615–1619 (1993)

Railway Corridor Monitoring Using Deep Drone Vision

Sushanth Ikshwaku, Amrutha Srinivasan, Ashley Varghese and Jayavardhana Gubbi

Abstract The emergence of low-cost commercial drones fitted with a camera are ideal platforms for remotely monitoring critical assets such as railway corridor. The proposed system employs drones to automate and make the process efficient. In this paper, a railway monitoring system capable of detection and classification of various railway-related infrastructures such as lines, ballast, anchors, sleepers and fasteners using visual images captured by a drone is proposed. The first stage of classification uses a deep network that helps in qualifying the presence of track in a given frame. The second stage helps in classification of objects within a frame for further analysis. Two different deep architectures are used in classification of railway infrastructure—the first for offline analysis that uses transfer learning using a pre-trained GoogLeNet model and the second approach that uses a new architecture for embedded implementation. Transfer learning results in an overall f-score of 89%, and the new architecture results in an overall f-score of 81% with at least 10× reduction in parameters.

Keywords Rail track inspection · Convolutional neural network · Drones Transfer learning

S. Ikshwaku and A. Srinivasan were affiliated to BMS College of Engineering when the reported work was undertaken. Both authors contributed equally to this work.

S. Ikshwaku · A. Srinivasan
BMS College of Engineering, Bengaluru, India

A. Varghese (✉) · J. Gubbi (✉)
Embedded Systems and Robotics, TCS Research, Bengaluru, India
e-mail: ashley.varghese@tcs.com

J. Gubbi
e-mail: j.gubbi@tcs.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_28

1 Introduction

Railways played a major role in industrial revolution leading to economic prosperity of the current developed world. The modern railways are equally important and the recent efforts to join Asia and Europe through vast railway network is an indication of its significance. They not only bring people closer but inland freight movement becomes highly efficient as the sea routes invariably take longer time and are getting congested. As with any critical infrastructure, the railway corridor should be managed efficiently. The current state of the art includes manual surveys at regular intervals along the corridor. Although the system seems to work, there is scope for inefficiency and sabotage, specifically in developing countries. As larger networks become a reality, faster and more modern trains will become a norm. This will generate enormous potential for automatic monitoring system that has the ability to provide information as required and generate alerts depending on the state of the infrastructure.

Components of the railway track tend to contract in lower temperatures and expand in warmer temperatures. Ballast is constantly crushed when a train runs over it. Uneven expansion causes buckling, which is one of the many phenomena that result in derailment of trains. Sabotage of rails is also a another major concern in maintaining the safety of the railways. Proper surveillance and maintenance of such an expansive network of railway lines is tedious and complex, but also extremely necessary. Currently, railway lines are manually monitored on a periodic basis. People are employed to walk along tracks to identify missing or damaged rail components. The monotony associated with the job, extremely unsafe working conditions as well as the specific skill set and experience they require make it very difficult. Decisions made are highly prone to delays and human error. Therefore, the need arises for a system that can run independent of such operating constraints and perform at high efficiency irrespective of weather conditions.

Automating the process using drones would provide a safer, more robust and efficient way to monitor railway infrastructure. The advancements in drone technology and image analysis make it a viable solution to be used in this context. Nowadays drones are widely used for different applications in various areas, but their use in the monitoring of railway lines is still in a nascent stage. The camera attached to the drone enables it to record videos of the track as it flies over. Drone can follow rail lines autonomously using GPS or vision-based navigation, which is beyond the scope of this work. The videos are analysed using image processing techniques to detect anomalies. Usage of drones in this area increases the speed, efficiency and also autonomy of the system and requires less manual labour. This could purge costs and also risks pertaining to unsafe working environments.

In this work, a drone is employed to gather data of the railway track. Drone mounted camera is used to record RGB visuals while it is flying vertically above, along the direction of the track. The focus of the work is identification of the various parts of the railway track. The components are shown in Fig. 1 and include: *rails* (parallel bars laid on the ground to form a railway line), ballast (coarse stone or gravel

Fig. 1 Railway track components

laid on a railway track for stability), *crosstie* (beam that connects and supports rails), *fastener* (metal devices used to link rails to crossties on railway lines), *joint plate* (metal devices used to link rails on railway lines) and *anchors* (metal device used to support the railway track in order to resist longitudinal movement of rails under traffic). In this paper, we propose a deep convolutional neural network (CNN) architecture for detection of components on a railway line. Two different approaches are proposed: firstly, the network uses fine tuning of *Inception v3* pre-trained model, and another is a shallow CNN model. *Inception v3* is exemplary of the recent advancements of deep CNNs in image classification. However, it is pertinent to understand that with such deep networks comes a high computational complexity. In the context of drones, there is a limitation on the computational capabilities. Therefore, a lightweight deep model whose performance is comparable to that of the more complex networks would definitely be a better option for deploying it on drones. In this paper, we compare the performance of the transfer learning model using *Inception v3* and the proposed not-so-deep model. The discussions are restricted to detection and classification of components in railway corridor that forms the basis for critical infrastructure monitoring.

2 Related Work

The use of drones in the railway surveillance context is still new and being experimented in many countries. Flammini et al. [1] provide a survey of the current drone technology and their application in automating railway surveillance. It also highlights current technology and technical issues associated with drones. Visual surveillance covered by fixed cameras is not feasible as thousands are required to cover the vast railway network, and processing these videos manually is not possible. They support the use of drones with a mounted camera for surveillance with automated video content analysis. The work expands on certain applications of drones in railways like

post-disaster inspection, railway depot surveillance and rail line inspection. They discuss various collision detection and avoidance algorithms that are capable of running on the on-board GPU. They conclude by noting the improvements of on-board GPU resulting in the implementation of complex algorithms and the increased payload capacity of drones, which enables more on-board sensors for better working of the operating algorithms. The use of deep learning and the lightweight model is motivated by their comprehensive study where we perceive the use of full network can be done offline on the base station but the drone can have a lightweight detection algorithm for better surveillance.

Chellappa et al. [2] propose a fastener detection method using the traditional approach of extracting histogram of oriented gradient (HoG) features and using a combination of linear Support Vector Machines (SVM) classifier to detect anomaly in the fastener. Trinh et al. [3] propose a system that uses real time vision-based inspection by recording videos through four cameras mounted on a hi-rail inspection vehicle that provides a view of the left field, left gauge, right field and right gauge, respectively. It can detect the tie plate, tie, anchor and other components found on the railway track. The aforementioned works use computationally complex approaches, which are not viable on drones. As rail monitoring using vision is a nascent area, it is to be noted that not many papers are available.

Krizhevsky et al. [4] propose a very advanced deep convolutional neural network (DCNN) of its time, that is trained with 1.2 million high-resolution images for 1000 class object classification. It could be said that this work was one of the pioneering effort for showcasing the use of deep convolutional neural networks for image classification. *Inception v3*, one such model pre-trained on the ImageNet data set, has been used in multiple applications using transfer learning for image classification. Gibert et al. [5] states the advantage of using deep convolutional networks in overcoming constraints imposed by limitations in the data set, i.e. to achieve scalability and proposes a railway track inspection system that uses a multi-task learning framework to find the fastener location and then classify it as good, bad or missing. In the railway component classification scenario, we need very fine-grained classification, to detect the smaller components effectively. For fine-grained classification, some modification of these networks is required, especially in cases of context-specific classification. This poses a challenge in our problem statement.

3 Deep Learning for Drone Image Analysis

The flow chart of proposed architecture is shown in Fig. 2. The overall approach is divided into two stages: first stage involves detection of presence of the track within a given frame, and the second stage is classification of the railway components. The frame with a track (obtained due to first stage) is divided into 32×32 pixel blocks (empirically chosen), and each patch forms an input into the component classification module. In this stage, the components of the track in the image are classified, and connected component-based post-processing is performed to reduce false alarms.

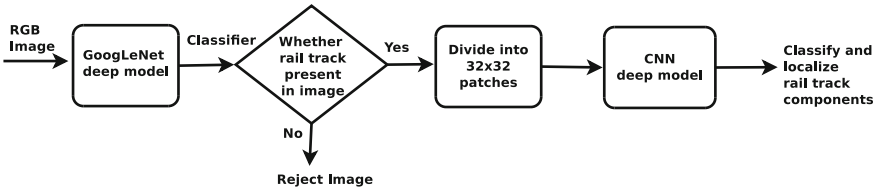


Fig. 2 Flow chart of proposed architecture

3.1 Convolutional Neural Network (CNN) for Detection of Rail Track

The purpose of rail track detection is to filter the images that have no tracks. The qualification is required as the detection and assessment of track depends on the distance from which the image is gathered. Further, the drift and other drone navigation issues may contain frames that are not correctly focused. The camera gimbal will alleviate some of the issues but to ensure that the components are easily classifiable in the second stage, frame qualification is introduced. In order to qualify the frames, transfer learning using a pre-trained ImageNet model is used. The last layers of the model are replaced with a with a global average pooling layer, a fully connected layer with *relu* activation and a *softmax* layer. The top layers are trained to classify an input image as Track or No Track. The input to the model is 224×224 greyscale image. The kernel size was chosen to be 5×5, and global average pooling was chosen to reduce the dimensionality of the feature map. The global average pooling layer has an input shape of $5 \times 5 \times 2048$ and resizes the feature map to 2048 dimensional vector which is input to the fully connected layer having a hidden layer with two neurons. The *softmax* layer provides the binary classification detecting the presence of a rail track. If a rail track is detected, the image is accepted for component classification, else the image is rejected. The architecture diagram for rail track classifiaction is shown in Fig. 3.

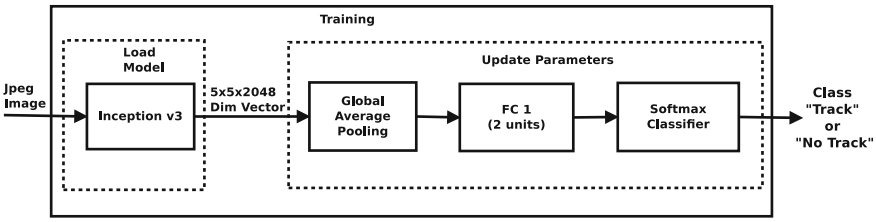


Fig. 3 Architecture diagram for rail track classification

3.2 Multiclass Classification of Rail Track Components

DCNN is used for the classification of the various components of the rail track. Two approaches are considered: (a) transfer learning using a state-of-the-art pre-trained deep CNN; and (b) develop a not-so-deep CNN for the component classification, which is more favourable in real-time analysis on-board a drone.

Pre-trained model In the transfer learning approach, just like the first stage, a pre-trained CNN model with last two layers is retrained to classify the components of the rail track. The architecture diagram for transfer learning approach for track component detection is shown in Fig. 4. We use *Inception v3* model trained on ImageNet 2012 data. This model has the required feature set to distinguish between over 1000 categories. The GoogLeNet (with Inception v3) is a 48-layer architecture, and this pre-trained model without its top layers is used for identifying the components of the rail track by adding a global average pooling layer and a fully connected layer. The pre-trained network weights can be transferred to the new model to distinguish the components of the rail track as the components of interest have similarities to the trained data set. For instance, the *fastener* is similar to a screw, the *ballast* is similar to gravel used in construction material, and *crossties* are wooden planks. This type of model results in a significant faster training time as the whole network need not be trained for the new application. The input to the model is a 32×32 greyscale image patch. The choice of the image patch is selected based on the smallest component of the rail track. A larger input size resulted in more than one component being present in the image rendering the classification invalid. The pre-trained model outputs a $5 \times 5 \times 2048$ dimension vector, which is provided as input to the classification layer. The global average pooling layer is used to reduce the dimension vector to 2048. The fully connected layer with seven neurons uses *relu* activation, and the *softmax* layer provides the final classification as one of the seven components.

Proposed CNN architecture for the rail track component classification The proposed network consists of four convolutional layers followed by two fully connected layers and a final *softmax* layer. The architecture of the proposed model is as shown in Fig. 5. The filters of the first convolution layer learn simple edges, and the filters of the subsequent convolutional layers are winning combinations of the previous filters. This allows the network to learn complex spatial feature enabling CNNs to classify images.

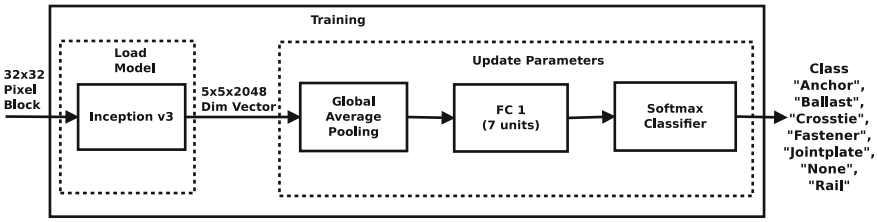


Fig. 4 Architecture diagram for transfer learning approach

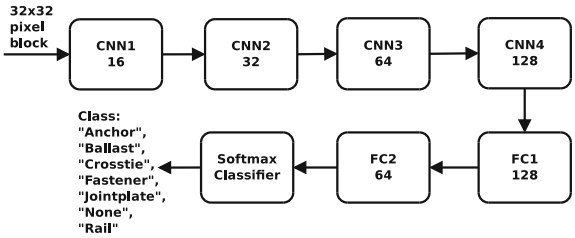


Fig. 5 Architecture diagram for shallow CNN for rail track component classification

The input to the CNN is a 32 × 32 greyscale image. The first convolution layer expands filters to 16 using a kernel of size 3 × 3 and one-third overlap with a stride length of two pixels. The same kernel size and stride length was maintained through all the convolutional layers in the model. The majority of filters of the first convolution layer learns simple straight edges when visualized. The second convolutional layer further expands the filters to 32; these filters learns a combination of the filters of the first layer, resulting in corners in many filters. The third convolutional layer expands the filters to 64. The last convolutional layer reduces the dimensionality to 128, and a flatten layer is used to reshape the feature maps. A *relu* activation layer is used after each convolutional layer for capturing nonlinearities. The rail track component classification is done by the *softmax* layer added at the end.

4 Experiments

Elastic Compute Cloud (EC2) instance on a Ubuntu-based system on Amazon Web Services with in-built 2.3 GHz Intel Xeon E5-2686 v4 processor. The model was built using Keras [6], a high-level neural network API library (with Tensorflow [7], a deep learning library that provided the backend for Keras) and Python 3.5 that supports the aforementioned libraries. The data set was collected by taking videos from a rail road vehicle on an 8 megapixel camera. These videos were then divided into image frames. We used a data set of 2017 images, randomly divided as training, validation and test data at the ratio of 6:3:1. The images from different view angle can be incorporated through data augmentation, which is not done for this paper. The resolution of images used for the data set was 256 × 256 in stage 1. For the second stage, only the lower half of the image was taken under consideration throughout as the smaller components were in focus and hence provided better clarity. We used LabelMe [8] to annotate the data. Only the lower half of the images were annotated that put a constraint on visibility of small components.

Table 1 Track classifier performance evaluation for various optimizers. All values in percentage

Samples	Optimizer	Precision	Recall	f-score
Epochs = 10,000 Samples per epoch = 200 Validation samples = 200	Stochastic gradient descent	95.76	95.76	96.58
	Root mean square propagation	97.43	98.27	97.85
	Adagrad	97.36	95.68	96.52
	Adadelta	98.27	98.27	96.27
	Adamax	97.39	96.55	96.96
	Nadam	94.26	99.13	96.63

4.1 Parameter Tuning

The first step of our model was to classify an image into two categories: Track or No Track This model was trained using all the images generated in data set for the Track class and images belong to No Track class consist of random objects and elements, some that were far away in terms of resemblance to a track such as flowers and animals to some that were closer in resemblance such as running track, roads and conveyor belts for the No Track class. The parameter optimization was performed for choosing the best optimizer. The performance metric for different optimizer is given in Table 1. The Root Mean Square Propagation (RMSPROP) optimizer was preferred for the model due to its higher f-score. The similar f-scores among various optimizers validate the transfer learning approach using pre-trained deep CNN. Further, as it will be evident from the result, the choice of parameters ensured that the classifier has very high sensitivity ensuring accurate recognition of frames with track and rejecting the rest.

All images that were classified as *Track* in the previous stage are subjected as input at this stage. Each image frame was divided into 32×32 pixel blocks. The annotated data from *LabelMe* were used as training examples for this model. The output of this stage has seven classes, one each for the six railway infrastructure components and one for none of them. First, the transfer learning approach was used to train the Inception v3 model. During training, a batch of 800 images and 2000 iterations was used. In the second approach, the same 32×32 pixel blocks were used to train the model. The CNN was experimented with different architectures, ranging in depth from three layers to ten layers. A CNN model with four layers deep provided the best results and hence has been used in our architecture. The stride length of the filters was varied from one to three. Stride length of two emerged as the most optimum. Regularization mechanisms like dropout layer and pooling layer were also used but failed to provide any improvement in the classification. During training, a batch size of 200 and 5000 iterations of the images was used.

Table 2 Component classifier performance evaluation for various optimizers. All values in percentage

Samples	Optimizer	Precision	Recall	f-score
Epochs = 20 Samples per epoch = 100 Validation samples = 50	Stochastic gradient descent	50.23	55.03	40.66
	Root mean square propagation	40.66	59.40	34.68
	Adagrad	47.67	59.25	47.67
	Adam	47.00	54.48	42.22
	Adamax	37.05	44.61	30.09
	Nadam	41.62	49.92	39.38
Epochs = 10,000 Samples per epoch = 150 Validation samples = 100	Stochastic gradient descent	61.89	69.59	64.13
	Root mean square	58.56	70.49	62.05
Epochs = 10,000 Samples per epoch = 500 Validation samples = 100	Root mean square	54.20	54.83	49.51
	Stochastic gradient descent	0	0	0
Epochs = 2000 Samples per epoch = 800 Validation samples = 100	Root mean square propagation	58.80	63.09	63.09
	Adagrad	64.16	68.26	65.16
	Adadelta	60.41	63.52	60.86

The component classifier is experimented with different optimizers as in the earlier case. The classifier performance metric for various optimizer is shown in Table 2. Among the optimizers tested, Stochastic Gradient Descent (SGD) failed to converge and resulted with the lowest accuracy. Adagrad optimizer provided the best f-score for stage 2 and was chosen in all our experiments. Among the various values of epoch and samples per epoch used, epoch values in the range of 2000 and sample values in the range of 800 provided highest accuracy.

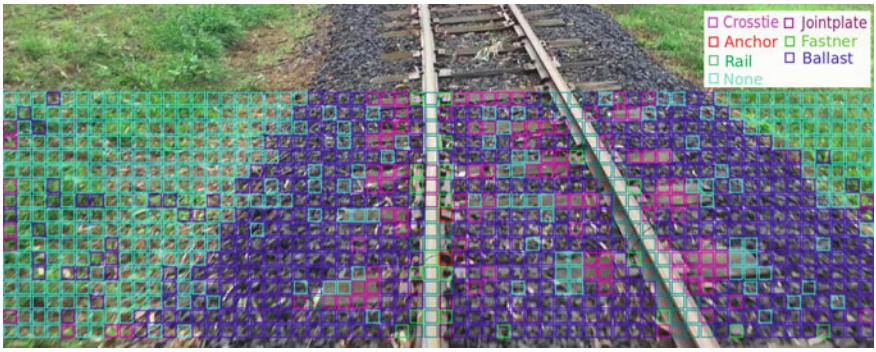


Fig. 6 Railway corridor component classification result

5 Results and Discussions

The trained model is tested with a set of test images. The Track–No Track classifier trained using RMSPROP optimizer classified the images with an f-score of 97.85%. The component classifier generated results as shown in Fig. 6. A sliding window of 32×32 is moved over the image for inferencing. Each class is represented with different colour of rectangle as shown in Fig. 6.

The performance evaluation of the proposed approach table comparing the performance of pre-trained model and the proposed CNN approaches is given in Table 3. A f-score of 93.72% is achieved for rail detection using proposed architecture against 89.79% obtained using transfer learning. Although the accuracy of the model is less for few components compared to the transfer learning model, the proposed CNN architecture is more practical in real-time processing using restricted computing resources.

A post-processing is performed on deep CNN output to improve the performance. Table 4 gives the result of rail track component classification after post-processing. As part of post-processing, several conditions for thresholding have been taken into

Table 3 Performance evaluation before post-processing. All values in percentages

Component	Pre-trained model			Proposed deep architecture		
	Precision	Recall	f-score	Precision	Recall	f-score
Anchor	5.6	77.78	10.45	78.94	83.33	81.08
Ballast	27.59	94.15	42.68	26.30	95.90	41.28
Crosstie	59.56	69.98	64.35	22.60	77.05	34.95
Fastener	22.45	83.02	35.34	25.58	83.02	39.11
Joint plate	31.44	85.91	46.04	67.5	78.26	72.48
Rail	96.05	84.30	89.79	91.94	95.56	93.72

Table 4 Performance evaluation after post-processing. All values in percentages

	Pre-trained model			Proposed deep architecture		
	Precision	Recall	f-score	Precision	Recall	f-score
Anchor	25.49	72.22	37.68	82.35	77.77	79.99
Ballast	55.55	95.03	70.12	30.67	96.49	46.5
Crosstie	86.08	49.70	63.02	60.13	56.13	58.06
Fastener	29.57	64.15	40.48	32.17	69.81	44.05
Joint plate	34.27	85.92	48.99	67.5	78.26	72.48
Rail	96.05	84.30	89.79	91.94	95.56	93.72

account. A single patch cannot appear as a line or ballast in isolation as some of the surrounding patches have to belong to the same class. Anchors and fasteners must be in close proximity to a joint plate or rail line. When an isolated component patch is surrounded by none, it is marked as none. After post-processing, the proposed model performed much better in terms of precision, recall and f-score, as compared to the transfer learning model. Notably, our model is able to detect the smallest component the anchor with far more accuracy than the transfer learning model. In the case of ballast, the proposed model could not learn the texture differences between the none class and ballast. Due to this there is no significant improvement in accuracy even after post-processing. The final f-scores for anchor, fastener, joint plate and the rail are higher than that of the transfer learning model.

The future work includes defect detection in railway corridor, and this paper proposes a method for detection and classification of components (which forms the fundamental step in inspection and navigation) in a railway corridor. The data set can be further improved to accommodate for varying degrees of illumination during different times of the day. It is important to note that the simplicity and short classification time of our model is better in comparison to the transfer learning model. In addition, a small increase in performance measure is achieved. The transfer learning model is 48 layers deep with a total parameter count of 21 million, whereas the proposed model has around 0.1 million parameters. In addition, the classification time is much faster using the proposed model. The pre-trained model takes 79.7905 ms approximately for classification, and our model takes 0.627 ms. The proposed model is more practical and deployable in drones having restricted computing power and where real-time analysis is required.

6 Conclusion

The aim of this work was to design a railway component classification system that can be deployed on a drone, i.e. one that requires less processing, and can perform faster than the existing deep models using transfer learning. The proposed framework

is capable to detecting and classification of various railway-related infrastructures such as lines, ballast, anchors, sleepers and fasteners using visual images captured by drone. A deep network is used for checking the presence of track in a given frame and achieved an f-score of 97.85%. Two different deep architectures are used in classification of railway infrastructure. The first is transfer learning and the second approach that uses a new architecture for embedded implementation. A post-processing is performed for improving the performance. The pre-trained model was trained using GoogLeNet, and an overall f-score of 89% was achieved. The proposed CNN architecture consisting of four convolutional layers followed by two fully connected layers, achieved an overall f-score of 81%. Further, for rail lines, which is the most critical component, an accuracy of 93% was obtained using the proposed not-so-deep network compared to 89% achieved by transfer learning model. The classification time and number of parameters of our proposed model are both lesser than that of the pre-trained model, making it a much more viable solution to deploy on drones. With a small compromise on the f-score, the proposed architecture outperforms the transfer learning model in terms of number of parameters and classification time. This would be a sensible solution in a scenario where smaller processors and real time analysis are concerned.

References

1. Flammini, F., Naddei, R., Pragliola, C., Smarra, G.: Towards automated drone surveillance in railways: state-of-the-art and future directions. In: International Conference on Advanced Concepts for Intelligent Vision Systems, pp. 336–348. Springer International Publishing, Oct (2016)
2. Gibert, X., Patel, V.M., Chellappa, R.: Robust fastener detection for autonomous visual railway track inspection. In: 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 694–701. IEEE, Jan (2015)
3. Li, Y., Trinh, H., Haas, N., Otto, C., Pankanti, S.: Rail component detection, optimization, and assessment for automatic rail track inspection. IEEE Trans. Intell. Transp. Syst. **15**(2), 760–770 (2014)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Adv. Neural Inf. Process. Syst. 1097–1105 (2012)
5. Gibert, X., Patel, V.M., Chellappa, R.: Deep multitask learning for railway track inspection. IEEE Trans. Intell. Transp. Syst. **18**(1), 153–164 (2017)
6. Keras: The Python Deep Learning Library. https://keras.io/
7. Tensorflow: An open-source software library for machine intelligence. https://www.tensorflow.org/
8. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Int. J. Comput. Vis. **77**(1), 157–173 (2008)

SURF–MSER Based 3D Mapping Using RGB-D Camera on Automated Vehicle

Nishchal K. Verma, Aquib Mustafa, Narendra Kumar Dhar and Vibhav Sarraf

Abstract 3D map generation of unknown environment is one of the challenging tasks in the field of localization and mapping. Navigation of automated vehicles in unstructured environment faces such problem. This paper presents 3D map generation with use of RGB-D camera placed on rotating platform mounted on vehicle. This arrangement enables 360° field of view for successive capturing of frames. SURF–MSER algorithm is proposed in this paper for feature detection and matching between the frames. The RANSAC and RANSAC-ICP algorithms are further implemented for estimation of transformation matrix between frames for point cloud generation. Loop closure detection is then performed with K-dimensional tree to establish complete 360° scene of environment. The developed 3D map of environment is globally optimized with the use of sparse bundle adjustment algorithm. Results obtained for two different laboratory setups have been shown which demonstrate the effectiveness of the proposed algorithm.

Keywords SURF · MSER · Mapping · Vision · RGB-D · Automated vehicle

1 Introduction

3D mapping has many applications in the field of mobile robotics like navigation, path planning, and object recognition [1–3]. The main components used for 3D mapping are finding spatial alignment between consecutive frames, detecting loop closures,

N. K. Verma · A. Mustafa · N. K. Dhar (✉) · V. Sarraf
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: nardhar@iitk.ac.in; dharnarendra@gmail.com

N. K. Verma
e-mail: nishchal.iitk@gmail.com

A. Mustafa
e-mail: aquibmustafa.225@gmail.com

V. Sarraf
e-mail: vidgaf@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_29

and globally optimizing the alignment for the whole map. With the introduction of the RGB-D cameras, 3D mapping has become a major research topic. Although color cameras can capture high-quality images and are very efficient for detecting loop closures, but extracting depth information from 2D images is a time-consuming task, and hence, these devices do not seem to be successful for this purpose. On the other hand, RGB-D cameras capture color images along with the depth value of each image pixel. The images obtained may have noises too. For noise removal there are several state-of-the-art algorithms [4, 5].

RGB-D cameras used to capture 3D information are either based on Active or Time of Flight [6] sensing to get the depth value for the pixels. The work in this paper used a Microsoft Kinect sensor [7] which includes a color camera, a depth camera, a tilting motor, and an accelerometer. The depth camera uses an IR sensor for depth sensing. A point cloud is generated by combining the color image and the depth image as captured by the RGB-D sensor.

In this paper, 3D maps of indoor environments have been constructed by capturing images from RGB-D camera. The main components used for 3D mapping are finding spatial alignment between consecutive frames, detecting loop closures, and globally optimizing the alignment for the whole map. The correspondences between the frames are found to align the two frames. This requires extraction of feature points from the images with further match between the consecutive images. Different types of features have been used for this purpose. SIFT features have the advantage of being invariant to scale and rotation but computing them takes more time. Inspired by SIFT are the SURF features [8] which are very fast to compute and have been found more robust. In this paper, SURF–MSER features are introduced which are combination of SURF and MSER [9].

After feature extraction, the RANSAC algorithm is applied on these feature points to find the transformation between these two frames [6, 10]. This transformation is improved by applying iterative closest point (ICP) algorithm [11] between the frames. For finding loop closures [12], a data structure is needed to match features between current frame and previous frames. A frequently used data structure for this purpose is vocabulary tree [13], but it suffers from perceptual aliasing problem. In this work, a simple KD tree [14] has been used for matching features as they are easy to implement and give better results. Keyframes [15] which are a subset of the frame already visited are found and then matched with each new frame to detect loop. However, by this method only the keyframes will appear in the loop closure pairs along with the frame matched. By making a simple modification in this algorithm without changing its time complexity loop pairs between the current frame and all the previous frames can be obtained. This gives more loop pairs and improves the obtained result from globally optimized map.

The final step in 3D mapping is global optimization. The sequential constraints as well as the loop closure constraints are included while performing this operation. Pose Graph Optimization using TORO [6] is one way of doing global optimization. The bundle adjustment in [6, 10, 16] has been found to have better performance than TORO. A joint 2D and 3D bundle adjustment approach has been used in this paper to solve the given problem.

The rest of the paper proceeds in the following manner. Section 2 introduces SURF–MSER features. Section 3 describes the RANSAC-ICP algorithm to align two frames. The algorithm for loop closure detection and global optimization is discussed in Sect. 4. The results are shown in Sect. 5. Finally, Sect. 6 concludes the paper.

2 SURF–MSER-Based Feature Points Detection and Matching

The alignment of any two frames is done by matching the points followed by transformation between the frames. To find the matching points, the feature points are first extracted from both the images followed by obtaining correspondences between the images. The state-of-art proposes different features for finding matches between two images. SIFT and SURF are generally used for this purpose. SIFT features use a Difference of Gaussians detector and a robust detector. The SIFT features are beneficial as they are invariant to scale and rotation. Another widely used features are MSER which are invariant to affine transformation of images' intensities. Among SIFT, SURF, and MSER, it has been found that SURF and MSER features are very fast to compute. The SURF points obtained are different from MSER points. These points are used for matching between the two frames. More the number of points, more is the probability to find matched points between two frames. The paper therefore uses combined SURF–MSER. These obtained features are used to find the transformation between the frames for alignment. The SURF, MSER, and SURF–MSER algorithms are discussed in next sections.

2.1 Speeded Up Robust Features

SURF is a robust local feature descriptor which is popularly used in vision applications such as object matching [17, 18] and image registration. SURF has been inspired by SIFT, and its algorithm works on same methodology as SIFT. SURF has been found to be faster and more robust than SIFT.

2.1.1 Interest Point Detection

SURF detection is done using scale-space theory [19] and Hessian matrix [8]. Hessian matrix is calculated to evaluate the local change around every pixel values. Only such pixel values are selected where the determinant is maximum. The Hessian matrix $\mathbf{H}_m(P_m, \sigma)$ for pixel position $P_m(x, y)$ and scale σ is determined as follows:

$$\mathbf{H}_m(P_m, \sigma) = \begin{bmatrix} L_{xx}(P_m, \sigma) & L_{xy}(P_m, \sigma) \\ L_{xy}(P_m, \sigma) & L_{yy}(P_m, \sigma) \end{bmatrix} \tag{1}$$

where $L(P_m, \sigma)$ is convoluted second-order derivative of the grayscale image I at point P_m. Gaussian is approximated using a box filter with a dimension of 9×9 and $\sigma = 1.2$. To make the convolution fast, integral image is used. The Hessian matrix determinant is computed using approximated convolutions D_{xx}, D_{xy} and D_{yy} is given by:

$$det(\mathbf{H}_{approx}) = D_{xx} D_{yy} - (w D_{xy})^2 \tag{2}$$

where filter response weight is delineated by w (generally chosen as 0.9). Then, the interest points are found at different scales by dividing the image into a number of octaves. Localization of interest points in the image is done by applying a non-maximal suppression in a $3 \times 3 \times 3$ neighborhood. Final interest points are determined by interpolating the maxima of Hessian matrix determinant in the image and scale space [20].

2.1.2 Local Neighborhood Descriptor

A descriptor is used to identify an interest point uniquely and robustly by describing the intensity distribution around that point. The dimension of a descriptor plays an important role in determining its computation complexity and matching efficiency. In SURF, the descriptor is first extracted by fixing the reproducible orientation around the feature point and then a square region aligned to the selected rotation is constructed.

2.2 Maximally Stable Extremal Regions

MSER is an affine invariant feature point detector and descriptor which is used for blob detection in images. The MSER algorithm [21] extracts co-variant regions from an image. These regions are then approximated by attaching elliptical frames to the MSERs which are further used to extract feature descriptors.

The pixels inside MSER are brighter or darker than all the pixels on boundary region and hence are called extremal regions. For MSER extraction, the image is taken through a series of thresholds of intensity from black to white. For each threshold, extremal regions are found by extracting connected components from the image at particular intensity. A threshold for an extremal region to be maximally stable is then decided by local minima of the relative growth of its squares. The stability of extremal region R_t is defined as,

$$\phi_1(R_t) = \frac{Ar(R_t)}{\frac{\partial Ar(R_t)}{\partial t}} \tag{3}$$

$Ar(R_t)$ denotes the area of R_t. A region is said to be stable if change in area is negligible with the change in threshold t. A region R_t is said to be maximally stable if $\phi_1(R_t)$ has a local maxima at t. These regions vary from fine to large structures. A region is approximated with an ellipse which is used as feature descriptors.

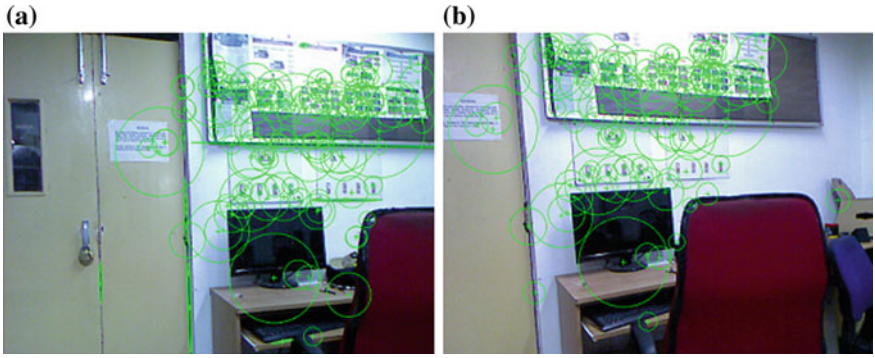


Fig. 1 SURF–MSER features extraction and matching **a** source image, **b** target image

2.3 SURF–MSER Features

SURF and MSER features extracted from an image are unified to increase the number of features. The resultant features are named as SURF–MSER features. These features are used to find association between two image frames for computation of transformation matrix. Figure 1 shows two images with SURF–MSER features matched between the images.

3 Image Acquisition and Frame Alignment

RGB-D camera captures the RGB image and the depth image per frame separately. Image frames are used to generate a point cloud. For RANSAC, these feature points need to be converted from pixel space to 3D space. Given $(\bar{u}, \bar{v}, d) \in \mathbf{R}^3$, where $\bar{u}$ and $\bar{v}$ are pixel coordinates and $\bar{d}$ is the depth of point in the point cloud. Equations (4), (5), and (6) transform these values into 3D coordinates, $(x, y, z) \in \mathbf{R}^3$.

$$x = \frac{\bar{u} - c_x}{f_x} \times d \qquad y = \frac{\bar{v} - c_y}{f_y} \times d \qquad z = \bar{d}$$

where f_x and f_y are the focal lengths in x- and y-direction and (c_x, c_y) is the center of image in pixels.

3.1 Aligning Two Frames Using RANSAC-ICP

Current frame is matched with the previous frame using SURF–MSER features to compute the relative transformation between them. The points which do not have depth values are excluded. Using the remaining associations, the RANSAC algorithm

finds the relative transform between these frames. The ICP algorithm is employed to improve the initially computed transformation. The algorithm for aligning two frames is given in Algorithm 1. The algorithm for RANSAC and ICP is discussed in the next sections.

Algorithm 1: Align Two Frames

Data: Frames F_1, F_2
Result: Relative Transformation T
$P_1 \leftarrow$ `ExtractFeatures(`F_1`)`
$P_2 \leftarrow$ `ExtractFeatures(`F_2`)`
$\mathbf{T}^* \leftarrow$ `RANSAC(`P_1, P_2`)`
$\mathbf{T} \leftarrow$ `ICP(`T^*, F_1, F_2`)`
return T

3.2 RANSAC

The relative transformation between pair of point clouds is determined using RANSAC by randomly choosing a minimum number of data points. SURF–MSER features are used to find the correspondences between the frames. The RANSAC is applied on these corresponding points after removing the points with invalid depth values. The square of the distance between the corresponding points in two frames is taken as the error metric [6].

$$\mathbf{T}^* = \min_{\mathbf{T}} \left(\frac{1}{|a_f|} \sum_{i \in a_f} w_i |\mathbf{T}(P_i^s) - P_i^t|^2 \right) \tag{4}$$

Here, a_f denotes the correspondences between the interest points in two frames. P_i^s and P_t^s are interest points in source frame and target frame. The interest points are in 3D space. w_i is the weight function, and $\mathbf{T}^*$ is the estimated transformation matrix. RANSAC algorithm takes 3D space coordinates of the matched feature points set as sample input and the distance error metric as fitting function and computes the transformation matrix between the pair of frames. It is done in two steps:

(1) A random sample of three pairs of interest points is selected. A optimized transformation matrix is found using only the points of this sample subset.

(2) The number of inliers is found by optimizing the error metric from the remaining interest points. For set which results in the maximum number of inliers, a more accurate transformation is found.

3.3 ICP

The ICP algorithm aims to find the rigid transformation between the pair of point clouds. A KD tree is used to accelerate the search of nearest neighbors. It is necessary for ICP algorithm that the frames are nearly aligned and have enough overlap,

otherwise it may converge at an incorrect local minimum. Given two point clouds A and B and the initial transformation matrix computed using RANSAC, the ICP algorithms have two basic steps:

(1) The associations between the point clouds are computed.

(2) The transformation matrix is found which minimizes the squared error between the associating points. Iterating the above two steps until convergence gives us the desired transformation.

Algorithm 2: Iterative Closest Point ICP

```
Data: Source Frame f_s, Target Frame f_t, Initial Transformation T_o
Result: Optimized Transformation T
i = 0
T(i) = T_o
while i < MaxIterations OR not converged do
    Matchings = ComputeClosestPoints(T(i), f_s, f_t)
    T(i+1) = LeastSquares(Matchings)
    if convergence then
        BREAK
    end
    i = i+1;
end
```

4 Loop Closure and Global Optimization

The camera when moves over long distances and comes back to an already visited location, the errors in the process of alignment during the journey of the camera gets added up and the result is an absurd map having same region present at different locations. This is called loop closure problem. To deal with this problem, first the loops are detected and then the map is globally optimized using bundle adjustment. These processes are described in the next subsections.

4.1 Loop Closure Detection

Loop closures need to be detected to find out the places that have been revisited. While generating 3D map of indoor environment, loop closures will certainly occur. Detecting these loop closures correctly, the errors that get accumulated during alignment process can be fixed. However, if incorrect loop closures are detected then it may lead to some serious errors in the map generation.

The process of loop closure detection mainly involves image comparison and matching. Bag of Words model is one such popular method used for this purpose. Using this model, one can represent a frame as a bag of visual words by quantizing feature descriptors which can be rapidly compared to other frames [22]. Although Bag of Words is a very powerful tool, it suffers from problem of perceptual aliasing

which occurs due to vector quantization. Another method for loop closure detection is to directly match features extracted from the images. Using a tree structure for this purpose proves to be an efficient method for feature matching. In this paper, matched features are directly extracted from the images using a tree structure to find loop closures. The SURF–MSER algorithm is used for finding matched feature points. The algorithm to detect loop closures is given in Algorithm 3 and is explained in the next subsections.

4.1.1 Feature Matching Using KD Tree

A KD tree is used to detect loop closures by matching features. The KD tree is a binary tree in which every node is a k-dimensional point. The KD tree reduces the search time complexity from linear to logarithmic. For each query feature vector, the KD tree returns its nearest neighbor along with the distance between them. They are considered as a correct match only if the distance between them is less than a given threshold.

4.1.2 Formation of KD Tree

The aim is to form a KD tree for all the frames that have been already visited so that each new frame can be matched with previously visited frames and loop closure can be found. However, it might be possible that a frame has some features that are common with other frames. To avoid building KD tree for all such frames which can be very expensive, only those frames are included which have very few common points between them. Such frames are called Keyframes. The simplest way to find keyframes is to consider every nth frame as the keyframe. Keyframes are determined on the basis of visual overlap. Generally, first frame is considered as a keyframe. For each new frame if the visual overlap between the current frame and the previous frame is less than the given threshold, then the current frame is considered as a new keyframe. To compute visual overlap between two frames, following equation is used.

$$D = \frac{2N}{N_1 + N_2} \tag{5}$$

N_1 and N_2 are feature points in images I_1 and I_2, respectively, and N is the number of matched features between the two images. The value of D determines the closeness of frames. The frames get farther with decrease in the value of D. For each new frame, D is computed for current frame and previous keyframe. If D is less than the threshold, then the current frame is considered as new keyframe. Every time a new keyframe is found, its features are added to the feature pool F which contains features of all the keyframes.

4.1.3 Finding Loop Pairs

For each new frame f_i, Algorithm 3 first extracts its features and then the function *MatchTree* finds the matches between the extracted features and pool of features F.

This function returns the indices of all the keyframes K whose features match with the features of current frame f_i. Out of keyframes K, only those frames are considered which are at a minimum distance from the frame f_i. In the keyframes K considered, each of the keyframe k_j has features in common with the frames E present in between itself and the keyframe preceding it. It might be possible that these interim frames E matching with the keyframe k_j can also match with the frame f_i. These frames E should also be considered for loop pairs. Using this method, number of loop pairs will increase which results in better bundle adjustment. For every keyframe, k_j in K frames E are determined. Frames E and k_j are matched with f_i and if the number of matching features is greater than a threshold than that frame and f_i is considered as a loop pair. Features associated with current frame is extracted and included to feature pool F. With the use of all determined loop pairs, final global optimization is performed using bundle adjustment.

Algorithm 3: Loop Closure Detection

```
Data: Images I
Result: Loop Closure Pairs LP
i = 1
Keyframes = [i]
Q(i) = ExtractFeatures(I(i))
F = Q(i)
for i ← 2 to |I| do
    Q(i) = ExtractFeatures(I(i))
    M = MatchTree(Q(i), F)
    for j ← 1 to |M| do
        if |M(j) − i| > minDist then
            for k ← Keyframes.Prev(M(j)) + 1 to M(j) do
                N = MatchFeatures(Q(k), Q(i))
                if N > minMatches then LP.Add((k, i))
            end
        end
    end
    l = Keyframes.Last
    N1 = |Q(l)|
    N2 = |Q(i)|
    N = MatchFeatures(Q(i), Q(l))
    D = N/(N1+N2)
    if D < maxOverlap then
        Keyframes.Add(i)
        F = F ⋃ Q(i)
        UpdateTree(F)
    end
end
return LP
```

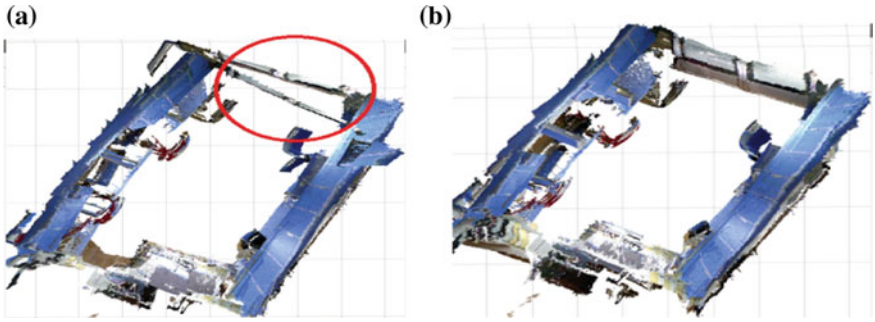


Fig. 2 Loop closure problem **a** without global optimization, **b** with global optimization

4.2 Bundle Adjustment for Global Optimization

The bundle adjustment is used to globally minimize the distance error between the feature points across all the frames. The method helps in dealing with loop closure problem. As the distance error of the feature points of the frames at the beginning and at the end of the loop get diminished, the final map looks quite similar to the actual indoor environment. The feature points are linked across multiple frames. The methodology adjusts the camera poses along with the estimated feature points' location. The initial camera poses are determined with the use of relative transformations. The bundle adjustment is fed with the initial camera poses and the constraints related to sequential frames and loop closure pairs. The constraints are information of locations of the feature points and the frames that have feature points in common. The bundle adjustment uses joint 2D and 3D objective function in Eq. (6) to provide refined camera pose estimates (Fig. 2).

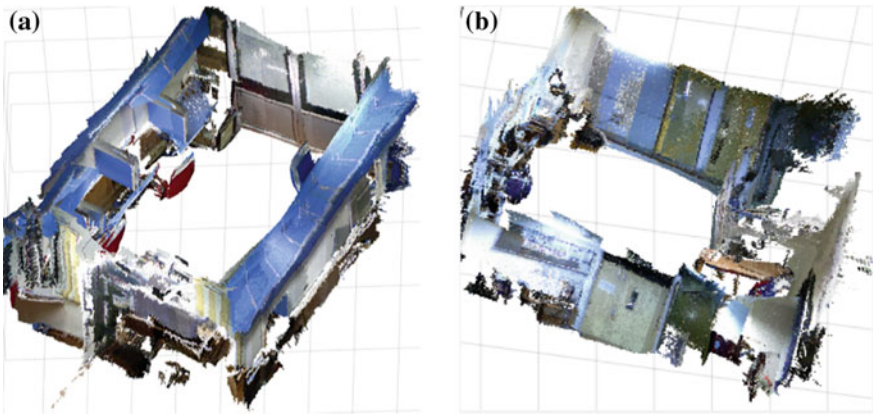


Fig. 3 Complete 3D map for **a** Laboratory 1 and **b** Laboratory 2

Fig. 4 Automated guided vehicle with RGB-D camera

Table 1 Comparison of SURF and SURF–MSER

	Matched features (average)	Time	Inliers
SURF	52.8	0.29	16.4
MSER	21	0.37	7.2
SURF–MSER	73.8	0.62	22.4

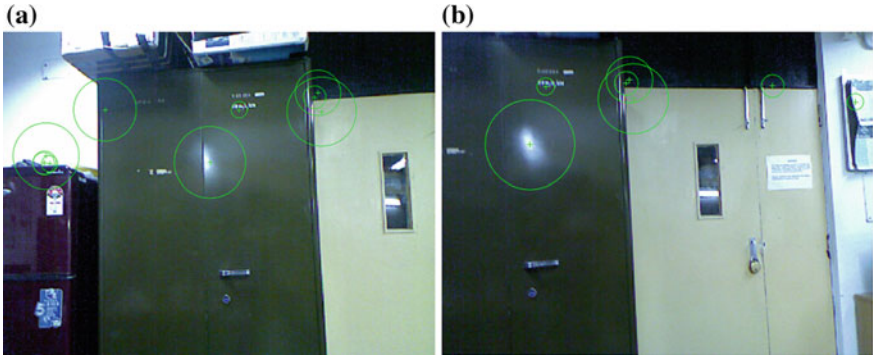


Fig. 5 SURF–MSER features extraction and matching in poor lighting **a** source image, **b** target image

$$min \sum_{c_i \in C} \sum_{p_j \in V(c_i)} (|(\bar{u}, \bar{v}) - Proj(\mathbf{T}_{c_i}(p_j))|^2 + \lambda |\bar{p_j} - \mathbf{T}_{c_i}(p_j)|^2) \tag{6}$$

where $Proj$ is a function which gives the 2D pixel location of a point when 3D location of point is given as input. $\mathbf{T}_{c_i}$ is the transformation matrix for the ith camera. p_j is the 3D location of a point with respect to camera c_i. (u, v) and p_j are the pixel values and 3D coordinates of the point p_j with respect to camera coordinate system, respectively.

5 Results and Discussion

The proposed algorithm is used to generate 3D map using RGB-D camera mounted on rotating platform of automated vehicle shown in Fig. 4. The 3D maps are generated for two different laboratory setups as shown in Fig. 3. The vehicle moves around the different parts of the laboratory while capturing frames.

The map for Laboratory 1 is generated using 44 image frames. Table 1 compares SURF, MSER, and SURF–MSER algorithms based on number of features matched between the frames, time taken per frame to align consecutive frames, and the number of inliers found per frame using RANSAC. The number of inliers is more in case of SURF–MSER, and hence, the transformation found is better than SURF or MSER alone. The results obtained using RANSAC are fairly good but those of RANSAC-ICP are better. However, RANSAC-ICP takes more time due to iterative closest point algorithm.

The map for Laboratory 2 was also taken the same way as Laboratory 1. However, the map of Laboratory 2 is not as accurate as Laboratory 1. The reason is poor lighting conditions which is one of the major problems in 3D mapping. The features extracted are very less, and they are poorly matched in poor lighting. This is shown in Fig. 5.

6 Conclusions

In this work, SURF–MSER algorithm is proposed for better extraction and matching of features between the frames for 3D mapping of indoor environment using a RGB-D camera. For aligning two consecutive frames, both RANSAC and combined RANSAC-ICP approaches are used. K-dimensional tree is proposed to be used in place of vocabulary tree which gives better results than existing algorithms in literature for loop closure detection. Joint 2D and 3D bundle adjustment method is used for global optimization of obtained 3D map. The proposed algorithm for generation of 3D map of indoor environment is effectively validated in different illumination

conditions, and the results for two different laboratories have been shown. This work can be used for generating 3D map for various mobile robotics applications such as navigation and path planning.

References

1. Verma, N.K., Dhar, N.K., Kar, A.K., Dev, R., Nawaz, S.S.F., Salour, A.: Internet of things based framework for trajectory tracking control. In: IEEE World Forum on Internet of Things, Reston (WFIOT), pp. 265–270. VA, USA, 12–14 Dec 2016
2. Kar, A.K., Dhar, N.K., Chandola, R., Nawaz, S.S.F., Verma, N.K.: Trajectory tracking by automated guided vehicle using GA optimized sliding mode control. In: IEEE International conference on Industrial and Information Systems (ICIIS), pp. 71–76. IIT Roorkee, India, 3–4 Dec 2016
3. Teja, B., Sharma, T., Verma, N.K., Sahoo, S.R.: Image dehazing for object recognition using faster RCNN. In: 2018 IEEE World Congress on Computational Intelligence (WCCI) (2018)
4. Dev, R., Verma, N.K.: Generalized fuzzy peer group for removal of mixed noise from color image. In: IEEE Signal Processing Letters (2018) (Early Access)
5. Singh, V., Dev, R., Dhar, N.K., Agrawal, P., Verma, N.K.: Adaptive type-2 fuzzy approach for filtering salt and pepper noise in grayscale images. In: IEEE Transactions on Fuzzy Systems (2018) (Early Access)
6. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-d mapping: using depth cameras for dense 3D modeling of indoor environments. In: Proceeding of International Symposium on Experimental Robotics (ISER) (2010)
7. Wan, Y., Wang, J., Hu, J., Song, T., Bai, Y., Ji, Z.: A Study in 3D reconstruction Using Kinect Sensor. In: 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)
8. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European Conference on Computer Vision (2006)
9. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference, pp. 384-396 (2002)
10. Xiao, J., Owens, A., Torralba, A.: Sun3D: a database of big spaces reconstructed using SFM and object labels. In: IEEE International Conference on Computer Vision (2013)
11. Sharp, G.C., Lee, S.W., Wehe, D.K.: ICP registration using invariant features. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2002)
12. Liu, Y., Zhang, H.: Indexing visual features: real-time loop closure detection using a tree structure. In: IEEE International Conference on Robotics and Automation (2012)
13. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
14. Bentley, J. L.: Multidimensional binary search trees used for associative searching. Commun. ACM (1975)
15. Zhang, H., Li, B., Yang, D.: Keyframe Detection for Appearance-Based Visual SLAM. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2010)
16. Fioraio, N., Di Stefano, L.: Joint detection, tracking and mapping by semantic bundle adjustment. In: CVPR (2013)
17. Verma, N.K., Goyal, A., Vardhan, A.H., Sevakula, R.K., Salour, A.: Object matching using speeded up robust features. In: 9th INNS Asia Pacific Symposium on Intelligent and Evolutionary Systems (2015)
18. Vardhan, A.H., Verma, N.K., Sevakula, R.K., Salour, A.: unsupervised approach for object matching using speeded up robust features. In: IEEE Applied Imagery Pattern Recognition Workshop, (AIPR) (2015)
19. Lindeberg, T.: Discrete scale-space theory and the scale-space primal sketch. Ph.D. thesis, Royal Institute of Technology (1991)

20. Verma, N. K., Nama, P., Kumar, G., Siddhant, A., Ocean, Raj, A., Dhar, N. K., Salour, A.: Vision based object follower automated guided vehicle using compressive tracking and stereo-vision. In: IEEE Bombay Section Symposium (2015)
21. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (MSER) tracking. In: CVPR (2006)
22. Verma, N.K., Kumar, G., Siddhant, A., Nama, P., Raj, A., Mustafa, A., Dhar, N.K.,Salour, A.: Vision based obstacle avoidance and recognition system. In: IEEE Workshop on Computational Intelligence (2015)

Rotation-Invariant Descriptor for Disparate Images Using Line Segments

Piyush Sahoo, Teena Sharma, Pooja Agrawal and Nishchal K. Verma

Abstract In recent years, computer vision applications have extended to a very wide range, which in turn encompasses a large variety of situational images and videos. This paper modifies the Duality Descriptor (DUDE), which uses line-point duality that provides simple consistent method of feature extraction. DUDE descriptor works very well for disparate image pairs, often outperforming most other methods with significantly less computation expenses. However, DUDE descriptor is not invariant to scale and rotation changes to the image, which is often vital for image processing in real-time scenarios. This paper modifies the existing DUDE descriptor, making it invariant to rotation to a certain degree. The experiment has been performed for some real-time images of objects to show the viability of the proposed descriptor. Herein, multilayered neural network is also used to verify the results in terms of percentage accuracy.

Keywords Feature detection · Feature descriptor · Duality Descriptor · DUDE Line segment detector · Multilayered neural network · MLNN

P. Sahoo · T. Sharma (✉) · P. Agrawal · N. K. Verma
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: tee.shar6@gmail.com

P. Sahoo
e-mail: psahoo6667@gmail.com

P. Agrawal
e-mail: pooja.iitd@gmail.com

N. K. Verma
e-mail: nishchal.iitk@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_30

1 Introduction

In recent years, automatic feature matching has become very robust due to advancement of feature detectors. In the literature, there are various feature detectors, such as speeded up robust features (SURF) [1] and scale-invariant feature transform (SIFT) [2]. However, these detectors work very well only under the condition of consistent lighting, same modality, same sensor, and many other restrictions. When there are changes to these ideal conditions, the performance of these feature detectors drops down significantly. As such, there arises a need to consider detectors based on features invariant to such changes, such as line segments, alongside the previous feature detectors.

Many such line segment detectors have been introduced over the years. They have achieved moderate success for image matching in case of disparate images. For example, the method proposed by Hauagge and Snavely [3] detects symmetric structures in images and uses those as the base of their detector. Similarly, the descriptor proposed by Bansal and Daniilidis [4] is based on the eigen-spectrum of the joint graph constructed from all pixels in the image. Duality Descriptor [5] is among the latest in the group of line segment detectors. This detector is based on a group of line segments concentrated around several points of interest, rather than all the line segments of the image as a whole. This method mimics the feature descriptors produced by SIFT and SURF, the difference being that it is based solely on line segments. All of them use a challenging dataset, which include symmetric shapes, and also have dramatic changes in lighting, time period modality, etc. All of these descriptors perform very well under the various conditions of disparate images. A lot of previous work has been done for speeded up robust features [6–12]. Most of them aim to remove the difficulties faced by such detectors, such as image disparity, textureless images, future image frames prediction [13–15]. Although they enjoy some success in making the descriptors robust to a certain degree to these features, the cost of that is a large increase in computational expenses. However, there are certain restrictions present while using these images. For example, the descriptor proposed in [3] requires the image to be symmetric, while the one in [4] requires extensive usage of time and memory for eigen-decomposition of such a huge matrix. DUDE does not suffer from such restrictions; however, it is not scale or rotation invariant. The focus of this paper is to achieve rotation invariance for the Duality Descriptor.

Herein, the descriptor design of the Duality Descriptor is modified slightly so as to make it invariant to rotation to a certain degree. As of now, it is still not invariant to rotation to all angles, but for a range of angles centered around multiples of 90° the descriptor is invariant to rotation changes even for disparate images. The method of formation of the feature descriptor differs from the original descriptor design, in an effort to make the descriptor invariant to rotation. The proposed feature descriptor is lightweight and requires less computation as compared to the existing line descriptors. Its performance is comparable to feature descriptors for the case of single image matching. Multilayered neural network is used for verification of proposed descriptor matching in terms of training and test accuracy.

The remainder is arranged as: Sect. 2 describes the existing DUDE descriptor. Section 3 explains the proposed modified descriptor. Section 4 shows results of the proposed descriptor. Section 5 concludes the paper.

2 Preliminaries

2.1 Duality Descriptor (DUDE)

Duality Descriptor was recently introduced in by Kwon et al. [5] The DUDE descriptor was proposed as a robust descriptor for disparate image matching based on line segments. This descriptor uses a 3D cylindrical histogram based on transformation of line segments to a dual space of points. DUDE was proposed as a descriptor which was supposed to be mainly used in the case of disparate images, i.e., images with dramatic appearance changes, such as different modalities, sensors, age, lighting. It was seen that DUDE performed significantly better than other line segment detectors

Fig. 1 Disparate image matching using DUDE [5]

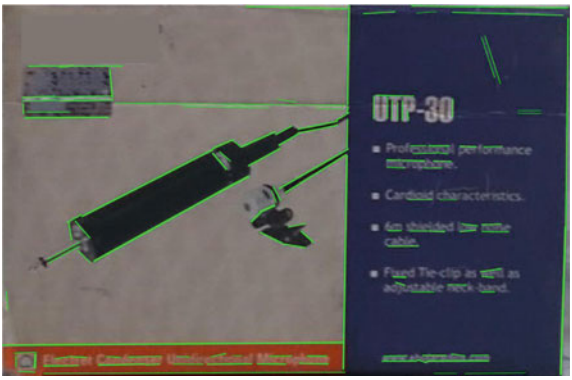


Fig. 2 Line segments in input image

Fig. 3 Line segments around a point of interest

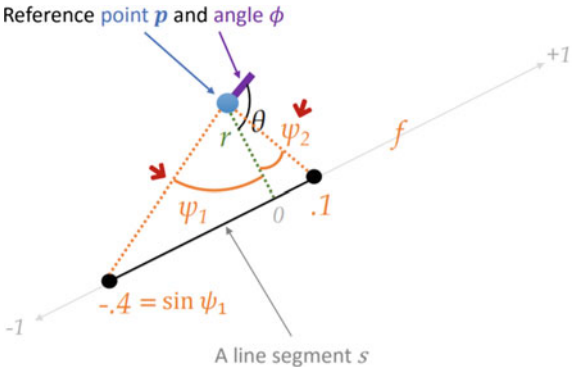


Fig. 4 Conversion of line segment to $[r, \theta, f_1, f_2]$ domain [5]

when used on a database set of disparate images. However, DUDE was not designed to be rotation or scale invariant. Since this descriptor is based on line segments [16], it is highly sensitive to variations to scale and rotation. Figure 1 shows the matching between a set of highly disparate images using the unmodified DUDE descriptor (Figs. 2 and 3).

Duality Descriptor (DUDE) is based on the concept of line-point duality, according to which this descriptor converts the line segments from an $[x_1, y_1, x_2, y_2]$ space to an $[r, \theta, f_1, f_2]$ space, where r is the distance of infinite line containing the line segment from the point of interest, θ is the orientation of aforementioned line, and f_1, f_2 symbolize the endpoints of the line segments. The descriptor formation can be seen in Fig. 4. This descriptor takes into consideration certain points of interest, around which the line segments are considered for forming the descriptor. For our

case, we have considered the SURF points as the reference points, since they are the most widely prevalent descriptors. It is to be noted that any other points, such as SIFT, corner points, can also be used without loss of information.

Typically, for any image matching descriptor, such as SIFT [17], the descriptor for a feature F_i is often defined in the form of $[x_i, y_i, s_i, \theta_i]$, which represent the location, scale, and orientation of the descriptor, respectively. DUDE descriptor is also defined in a similar fashion.

2.2 Multilayered Neural Network (MLNN)

Neural networks play a vital role for development of intelligent control systems. They are used extensively for system identification and controller parameterization. The neural network used for identification here is called multilayered neural network (MLNN). Here, two layers are used for recognition.

MLNN consists of an input layer, two hidden layers, and an output layer. The input layer consists of the training image datasets. Since there are four datasets, the output layer also consists of four outputs. These outputs are labeled using binary numbers for better identification. The neural network is trained on the training datasets, and then the network is then tested on the testing datasets for recognition.

3 Proposed Descriptor

This section describes the descriptor proposed in this paper, which is an extension of the descriptor in [5] by improving certain degree of rotation invariance.

3.1 Descriptor Design

The descriptor design is as follows. The chosen point of interest is selected as the center around which line segments are considered as part of the descriptor. The line segments which lie inside a circle of radius qs_i, where s_i is the scale of the chosen point, are identified. For each line segment, it is converted into the $[r, \theta, f_1, f_2]$ space, relative to the point at (x_i, y_i) and orientation θ_i. Then for each set of line segments, all the r, θ, f_1, f_2 are accumulated in a histogram, whose r, θ and (f_1, f_2) axes are divided uniformly into n_r, n_θ, and n_f bins. For the experiment presented in this paper, the values of n_r, n_θ, and n_f chosen are 5, 5, and 10, respectively.

Let the endpoints of a line segment be $q_1 = (x_1, y_1)$ and $q_2 = (x_2, y_2)$, and let the reference point be $p_i = (x_i, y_i)$ at an orientation of θ_i. Then, we have

$$r = \frac{\left| x_i * \frac{y_2 - y_1}{x_2 - x_1} - y_i + y_1 - x_1 * \frac{y_2 - y_1}{x_2 - x_1} \right|}{\sqrt{1 + \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2}} \tag{1}$$

$$\theta = tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right) - \theta_i \tag{2}$$

$$f_1 = sin \left(cos^{-1} \left(\frac{r}{|p_i - q_1|} \right) \right) \tag{3}$$

$$f_2 = sin \left(cos^{-1} \left(\frac{r}{|p_i - q_2|} \right) \right) \tag{4}$$

Furthermore, Algorithm 1 shows a concise algorithm of the proposed DUDE descriptor.

For the detection of the line segments used in the detector, the line segment detector (LSD) proposed by Gioi et al. [3] is used. The basic crux of LSD lies in detecting a set of pixels having gradient values within a tolerance of τ of each other and then using the region-growing method to complete the line segment. This line segment detector is linear time and gives sub-pixel accurate results. Figure 2 shows the line segments detected in our test image using LSD.

One underlying difficulty is that this descriptor is highly sensitive to the endpoints of the line segments. To counter this problem, the endpoints of each line segments are intentionally perturbed in a 3×3 region centered on the endpoint, followed by duplication of the line segment d times, by selecting an endpoint randomly from the perturbation area. This method makes the detector less sensitive to small changes to the endpoints. Figure 3 shows the resulting line segments around a single point of reference as an example when the perturbation is applied.

Another difficulty of most line segment detectors is the occurrence of segmentation of a single line segment into multiple parts, which can often lead to false results. By using line-point duality, DUDE is designed such that it is less affected by line segment disconnections, while simultaneously capturing geometric properties properly.

Algorithm 1: Proposed Descriptor

(1) Take input image I_m.
(2) $lines$ = Line Segments detected by LSD
(3) Left and right point of each line segment $i = (x_l, y_l)$, $i = (x_r, y_r)$
for i=1:n_{lines}
  $x_{l_{rand}}, y_{l_{rand}}, x_{r_{rand}}, y_{r_{rand}} = d$ random numbers in 3×3 matrix around x_l, y_l, x_r, y_l, respectively
  $lines = [lines, x_{l_{rand}}, x_{r_{rand}}, y_{l_{rand}}, y_{r_{rand}}]$
end
(4) Number of Points of Interest (POI) = n_{poi}
for i = 1:n_{poi}
  (x_i, y_i) = Location of POI, s_i = Scale of POI, θ_i = Orientation of POI, $q * s_i = r_0$(radius)
  for j = 1:n_{lines}
    (l_x, l_y), (r_x, r_y) =left and right intersection pt. of lines with circle (x_i, y_i) & r_0
  end
  for j=1:$length(lx)$
    l_1 = line segment joining (l_x, l_y) and (x_i, y_i)
    l_2 = line segment joining (r_x, r_y) and (x_i, y_i)
    l_3 = line segment joining (l_x, l_y) and (r_x, r_y)
    $r_j = \perp^r$ distance between (x_i, y_i) and l_3
    θ_j = angle between l_3 and θ
    $(f_1)_j, (f_2)_j$ = sin(angle between r_j & l_1 and r_j & l_2)
  end
  dude = $hist(r, \theta, f)$
end

4 Experimental Results

In this paper, test has been performed on an image, which is selected as the prototype, and then rotated that same image through various angles to confirm matching. So far, the descriptor shows robust matching for angles in the range of $0° \pm 10°$, $90° \pm 10°$, $180° \pm 10°$, and $270° \pm 10°$.

Figure 5 shows the matched features when the image is not rotated with respect to the prototype. As can be seen, almost all the corresponding features in both the images are matched.

Figure 6 shows the matching when the scene image is rotated by 5°. Here, the matches are reduced, but still sustain consistent matching of almost five features.

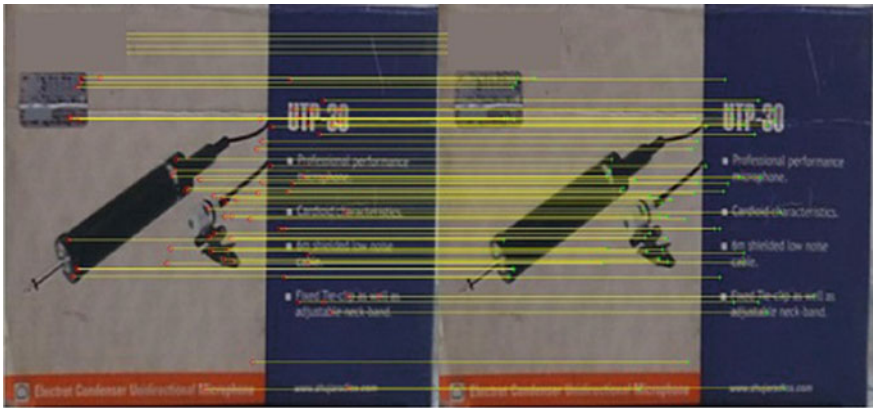


Fig. 5 Sample image 1—image matching for orientation angle of 0°

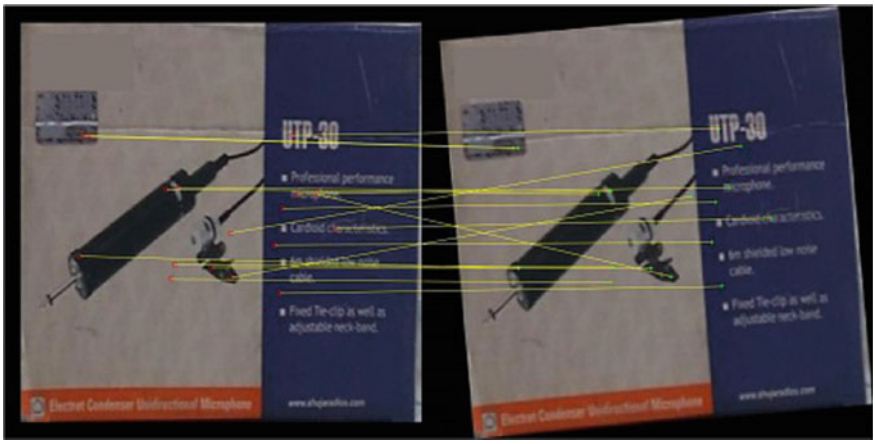


Fig. 6 Sample image 2—image matching for orientation angle of 5°

Figure 7 shows the matched features for scene image rotated by 10°. The number of matched features is further reduced, but still consistent.

Figure 8 shows the matched features for a rotation of 80°. The number and type of matched features are similar to the case of rotation by 10°.

Similarly, Fig. 9 shows the matched features for an angle of 85°, which is similar to the case of 5°.

Figure 10 shows the matched features for a rotation of 90°. Here, the number of matches is more as compared to other angles, though it is still less than the case of no rotation. Furthermore, the matches are consistent.

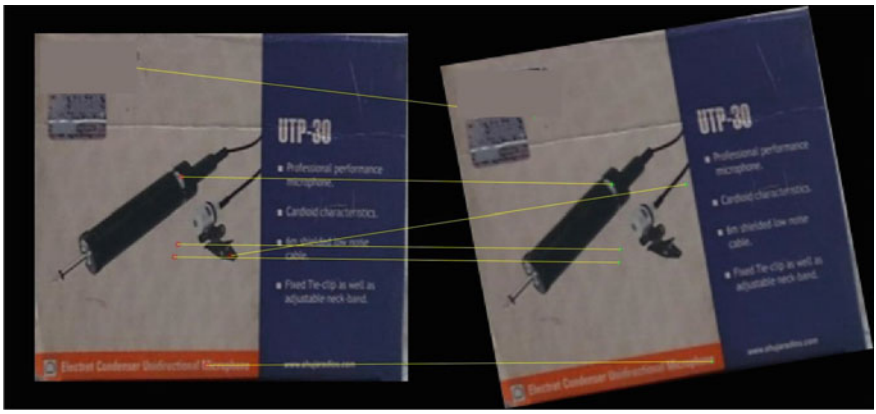


Fig. 7 Sample image 3—image matching for orientation angle of 10°

Fig. 8 Sample image 4—image matching for orientation angle of 80°

Figures 11 and 12 show the matched features for rotation angles of 95° and 100°, respectively. The number and types of matched features are similar to those of 5° and 10°, respectively.

Figures 13 and 14 show the matched feature descriptors for angles of 180° and 270°, respectively. These matches are similar to that in the case of 90°.

Figures 15, 16, 17, 18, 19 and 20 show the matched features for angles of 0° and 90° for three other sample images.

As can be seen in Table 1, the proposed descriptor outperforms the original descriptor, while having almost the same computational requirements. Table 1 shows the ratio of total matches to false matches, for both the cases of DUDE descriptor and proposed

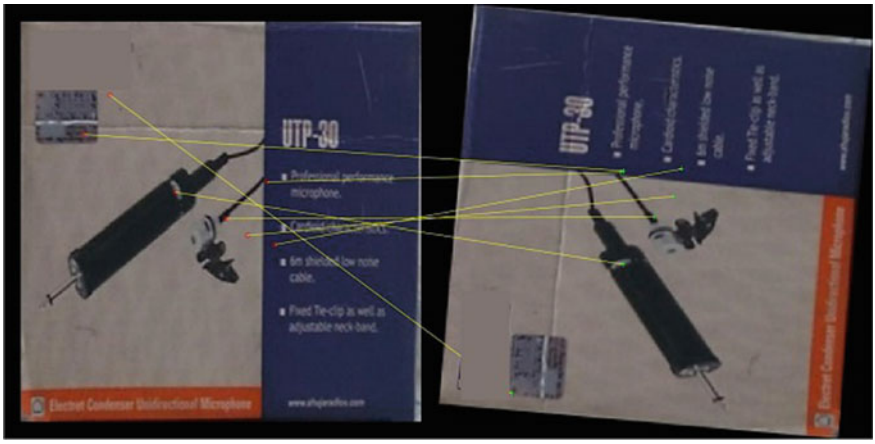


Fig. 9 Sample image 5—image matching for orientation angle of 85°

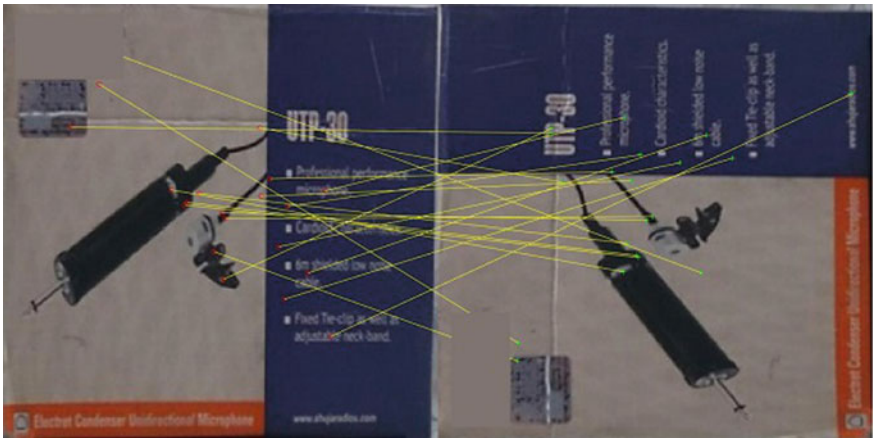


Fig. 10 Sample image 6—image matching for orientation angle of 90°

descriptor. In this table, the total number of matches, the number of false matches and the ratio of the total matches to the false matches for scene image, rotated at angles of 0°, 5°, 10°, 80°, 85°, 90°, 95°, 100°, 180°, and 270°, for the method proposed by Kwon et al. [5] and proposed method, are shown. As can be seen, the ratio of total to false matches is more for all angles in the proposed method as compared to that in [5]. Here, match ratio (MR) is defined as:

$$\text{Match Ratio (MR)} = \frac{\text{Total Matches (TM)}}{\text{False Matches (FM)}} \tag{5}$$

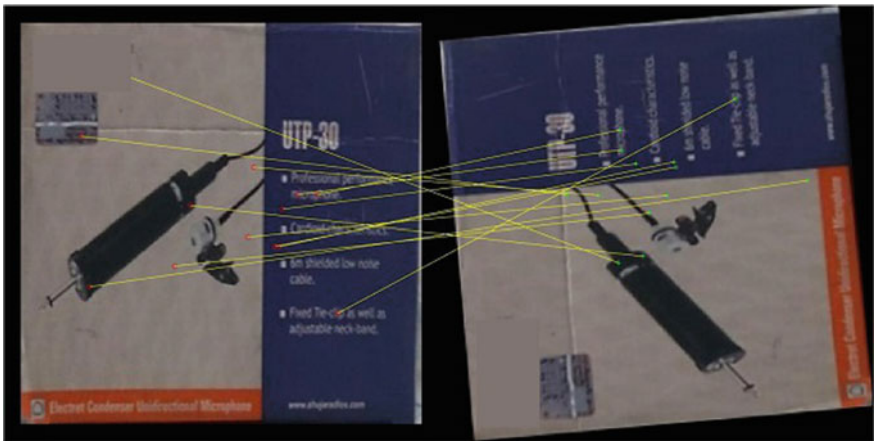


Fig. 11 Sample image 7—image matching for orientation angle of 95°

Fig. 12 Sample image 8—image matching for orientation angle of 100°

To find out if the image is matched correctly, the match ratio is used. In this case, it is assumed that as long as the number of false matches does not exceed half of the number of total matches, then a match is found. Thus, for MR ≥ 2, it is assumed that a correct match is found.

Table 2 shows the desired output, the actual output, the error, the MSE, and the percentage recognition for the results obtained from the test dataset. As can be seen, the percentage recognition accuracy is within acceptable ranges and thus feasible.
As can be seen, the number of matches gets progressively reduced as the angle of rotation increases. However, consistency is maintained. Furthermore, number of matches increase when the image is rotated by multiples of 90°. The reason for such a situation is that when the images are rotated, the originally single long line segment

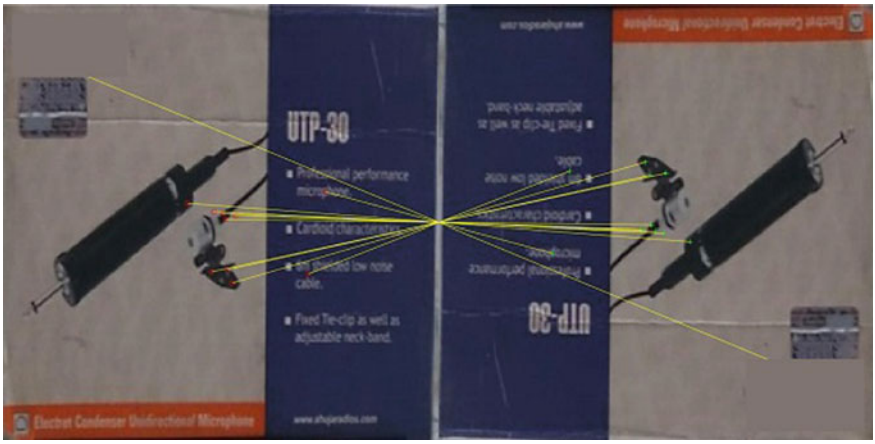


Fig. 13 Sample image 9—image matching for orientation angle of 180°

Fig. 14 Sample image 10—image matching for orientation angle of 270°

Fig. 15 Sample image 11—image matching for orientation angle of 0°

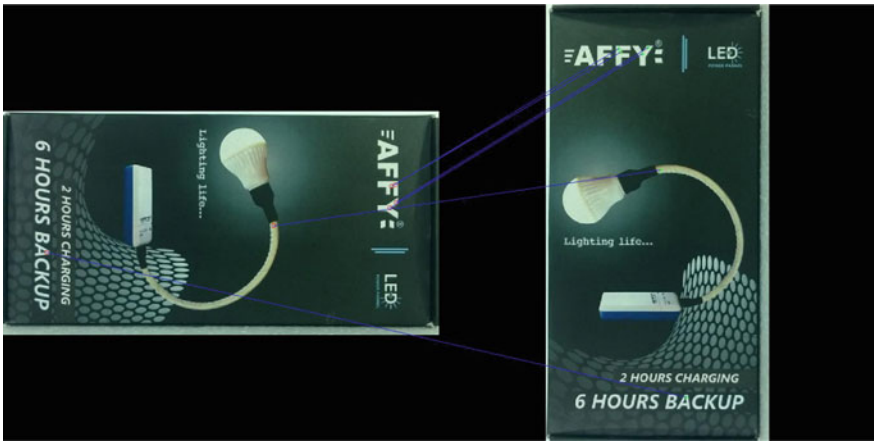


Fig. 16 Sample image 12—image matching for orientation angle of 90°

Fig. 17 Sample image 13—image matching for orientation angle of 0°

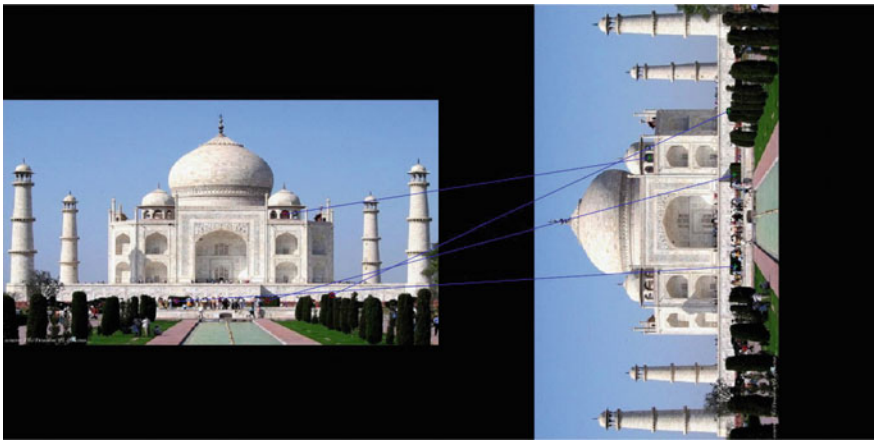


Fig. 18 Sample image 14—image matching for orientation angle of 90°

Fig. 19 Sample image 15—image matching for orientation angle of 0°

Fig. 20 Sample image 16—image matching for orientation angle of 90°

gets broken up into multiple short line segments, which in turn affects the descriptor formation.

Furthermore, the descriptor is also shown to be working for disparate images. Here, the matching between the same images, but both of them having a very large difference in contrast, color, and lighting, is shown. The case shown is one of the most common examples of disparate image matching required for real-life situations. The matching is shown for angles of 0°, 90°, 180° and 270°. Figures 21, 22, 23 and 24 show matching between disparate images when rotated at angles of 0°, 90°, 180°, and 270°, respectively.

Table 1 Match ratio for different orientation angles with respect to input image

Image orientation angles	Match ratio (MR) for Kwon et al. [5]			Match ratio (MR) for proposed method		
	Total matches (TM)	False matches (FM)	MR = $\frac{TM}{FM}$	Total matches (TM)	False matches (FM)	MR = $\frac{TM}{FM}$
Image 1 at 0° as shown in Fig. 5	74	2	37	75	1	75
Image 2 at 5° as shown in Fig. 6	6	4	1.5	17	2	8.5
Image 3 at 10° as shown in Fig. 7	0*	0*	N/A*	6	3	2
Image 4 at 80° as shown in Fig. 8	0*	0*	N/A*	9	5	1.8
Image 5 at 85° as shown in Fig. 9	0*	0*	N/A*	7	3	2.3
Image 6 at 90° as shown in Fig. 10	13	2	6.5	22	2	11
Image 7 at 95° as shown in Fig. 11	0*	0*	N/A*	13	6	2.2
Image 8 at 100° as shown in Fig. 12	0*	0*	N/A*	5	3	1.7
Image 9 at 180° as shown in Fig. 13	7	2	3.5	11	1	11
Image 10 at 270° as shown in Fig. 14	4	2	2	7	3	2.3
Image 11 at 0° as shown in Fig. 15	80	3	26.7	78	2	39
Image 12 at 90° as shown in Fig. 16	6	1	6	14	2	7
Image 13 at 0° as shown in Fig. 17	84	2	41	103	2	51.5
Image 14 at 90° as shown in Fig. 18	4	1	4	20	3	6.7
Image 15 at 0° as shown in Fig. 19	96	1	96	170	2	85
Image 16 at 90° as shown in Fig. 20	8	2	4	21	2	10.5

*No matches were found

Table 2 Comparison of percentage accuracy between DUDE [5] and proposed descriptor using MLNN

Dataset*	Expected output	MLNN output obtained		Training accuracy (%)		Test accuracy (%)	
		DUDE [5]	Proposed	DUDE [5]	Proposed	DUDE [5]	Proposed
Case 1	(0.01, 0.01)	(0.4308, 0.4308)	(0.196, 0.196)	22.80	82.31	15.84	74.32
Case 2	(0.01, 0.99)	(0.4308, 0.6344)	(0.0196, 0.8704)	51.67	85.19	22.36	68.40
Case 3	(0.99, 0.01)	(0.6344, 0.4308)	(0.8704, 0.196)	51.67	85.19	22.36	68.40
Case 4	(0.99, 0.99)	(0.6344, 0.6344)	(0.8704, 0.8704)	60.12	88.07	28.88	81.76

*Each dataset consists of ten images, from which six are used for training and four for testing

Fig. 21 Disparate image matching for orientation angle of 0°

In the case of matching of disparate images, the number of matches is naturally less when compared to the corresponding angles for the case of non-disparate images. This can be partly associated with the inconsistency of SURF features in case of disparate images, since the base of our descriptor formation is based on the feature point locations detected by SURF.

Fig. 22 Disparate image matching for orientation angle of 90°

Fig. 23 Disparate image matching for orientation angle of 180°

Fig. 24 Disparate image matching for orientation angle of 270°

5 Conclusions

In this paper, the DUDE descriptor has been modified so as to provide it a certain degree of invariance to rotation. Also, as can be seen, the rotation invariance persists even when the images are disparate with a very large difference in lighting and contrast. Further progress can be made to make the descriptor completely invariant to rotation, while also being invariant to scale changes and for multimodal images.

References

1. Bay, H., Tuytelaars, T., Gool, L. V.: SURF: speeded up robust features (2008)
2. Lowe, D.: SIFT: scale invariant feature transform (1999)
3. Hauagge, D.C., Snavely, N.: Image matching using local symmetry features. Comput. Vis. Pattern Recognit. 206–213 (2012)
4. Bansal, M., Daniilidis, K.: Joint spectral correspondence for disparate image matching. Comput. Vis. Pattern Recognit. **1**(2), 2802–2809 (2013)
5. Kwon, Y.P., Kim, H., Konjevod, G., McMains, S.: DUDE (duality descriptor): a robust descriptor for disparate images using line segment duality. In: IEEE International Conference on Image Processing (ICIP) (2016)
6. Agrawal, P., Sharma, T., Verma, N.K.: Supervised approach for object identification using speeded up robust features. Int. J. Adv. Intell. Paradig. (IJAIP) (2017)
7. Verma, N.K., Sharma, T., Sevakula, R.K., Salour, A.: Vision based object counting using speeded up robust features for inventory control. In: International conference on Computational Science and Computational Intelligence (CSCI'16), Las Vegas, Nevada, USA, 15–17 Dec 2016
8. Verma, N.K., Goyal, A., Vardhan, A.H., Sevakula, R.K., Salour, A.: Object matching using speeded up robust features. In: 19th INNS Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES15), pp. 415–427. Bangkok, Thailand (2015)

9. Vardhan, A.H., Verma, N.K., Sevakula, R.K., Salour, A.: Unsupervised approach for object matching using speeded up robust features. In: IEEE Applied Imagery Pattern Recognition Workshop, (AIPR), Washington, DC, USA, 13–15 Oct 2015
10. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
11. Verma, N.K., Sharma, T., Rajurkar, S.D., Salour, A.: Object identification for inventory management using convolutional neural network. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR). Washington DC, USA, 18–20 Oct (2016) (In Proceedings)
12. Verma, N.K., Goyal, A., Chaman, A., Sevakula, R.K.: Template matching for inventory management using fuzzy color histogram and spatial filters. In: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 317–322 (2015)
13. Verma, N.K., Sunny, N.K., Mishra, A.: Generation of future image frame using autoregressive model. In: IEEE Conference on Industrial Electronics and Applications, Auckland, pp. 171–176. New Zealand (2015)
14. Verma, N.K., Mishra, A.: Large displacement optical flow based image predictor model. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, Oct 2014
15. Verma, N.K., Singh, S.: Generation of future image frames using optical flow. In: Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, 23–25 Oct 2013
16. Gioi, R.G.V., Jakubowicz, J., Morel, J.-M., Randall, G.: LSD: a line segment detector. Image Process. Line **2** 35–55 (2012)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision (2004)

Object Detection and Recognition Using Small Labeled Datasets

Akhilesh Raj, Kanishk Gandhi, Bhanu Teja Nalla and Nishchal K. Verma

Abstract Object detection and recognition is a vibrant research area in the computer vision community. Several methods that came into scenario of object detection and recognition are expensive. This paper proposes another methodology for the same. We use selective search algorithm for providing region proposals where there is good chance of finding the object. The method is based on segmenting and eventually merging regions with good similarities. In this paper, we also propose a method for object recognition with a small labeled dataset for training. We use effective methods of unsupervised pre-training to effectively train the network. This paper tries to recognize objects using convolutional neural networks which are pre-trained using a sparse auto-encoder. The region proposals for the objects are forwarded to a convolutional neural network for feature extraction and finally into a fully connected layer for classification.

Keywords Object recognition · Selective search · Auto-encoders
Segmentation · CNN · RCNN

1 Introduction

The goal of object detection is to detect an object in a scene and try to recognize it from a pre-defined trained set of classes. This involves locating an object in a scene irrespective of orientation or scale. Images vary from each other hugely in terms of

A. Raj · K. Gandhi · B. T. Nalla (✉) · N. K. Verma
Indian Institute of Technology Kanpur, Kanpur 208016, India
e-mail: bhanutejanalla@gmail.com; bhanutn@iitk.ac.in

A. Raj
e-mail: akhileshraj91@gmail.com

K. Gandhi
e-mail: kanishk050296@gmail.com; kanishkg@iitk.ac.in

N. K. Verma
e-mail: nishchal@iitk.ac.in; nishchal.iitk@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_31

lighting, brightness, and contrast. The real-world detection and recognition of objects is often hard and computationally expensive [1]. In case of robotic challenges like target tracking using computer vision [2] future image frame generation [3–5], object identification and counting [6–10], human following, pick and place operations, the main part is object recognition and localization [11, 12]. The goal of an efficient algorithm is to detect and recognize the objects despite all the variances in the images while at the same time being computationally efficient. These variations in images call for a high requirement for a large labeled dataset of images to train a classification network. We try to find a way to reduce this requirement by performing pre-training images of the dataset in an unsupervised manner. Finding a large labeled dataset in real life is difficult and often a challenging and tedious task. Accumulating a large set of labeled images often takes a lot of time and effort. Minimizing this effort and to set upon to make an efficient algorithm to accurately detect objects in real life has been a growing need in today's world.

2 Previous Work

A study [13] has shown how learning good features can be essential while dealing with small labeled datasets. Matching the histograms of colors as described in Swain and Ballard's early work on color object recognition has brought new ways in which we tackle the problems in this domain [14]. The introduction of integral histogram by Porikli [15] has reduced the computational complexity of this algorithm. Interest point detection and the description of small-scale feature distribution are some drawbacks of these methods. Among descriptors, SIFT [16] has probably been one of the most widely used as it is invariant to rotation transforms and change in scale. This is due to the reason that SIFT captures a lot of information about spatial intensity patterns and also is robust as it is not affected much due to inaccuracies in localization.

The contributions of this paper are:

(1) An efficient object detection algorithm and recognition algorithm based on a short pre-training procedure.
(2) A combination of feature- and knowledge-based methodologies to segment image regions based on diverse set of similarity parameters that include color, texture, and fill is proposed.
(3) A methodology to forward the proposed region to a convolutional neural network (CNN) for feature extraction and finally to a fully connected layer for classification which is pre-trained using a sparse auto-encoder.

The paper explains the image segmentation in Sect. 3 followed by selective search algorithm for region proposals in Sect. 4. Object recognition using pre-trained convoluted neural network is described in Sect. 5 followed by a brief idea about the dataset used to test the algorithm in Sect. 6. The results and discussions are elaborated in Sects. 7 and 8 concludes the paper.

3 Image Segmentation

We use Felzenszwalb's graph-based image segmentation algorithm [17] to segment the image into its parts. It uses an efficient algorithm for determining the boundaries between the different regions of an image to segment using a graph-based representation of the image. The algorithm makes greedy decisions to produce segmentations that satisfy global properties. Selective search [18] works on merging these segmented regions based on various similarities, which include texture, color, size, and fill of proposed regions.

4 Object Detection: Selective Search

Popular algorithms in object detection use a sliding window approach to detect if a region has the probability of containing an object. This is computationally expensive [19] and requires a lot of time. Selective search [18] provides a novel approach in providing us with a quick, efficient, and yet accurate way of predicting if a region contains an object. The purpose of the algorithm is to get a bounding box around the object in the image. The selective search algorithm for object recognition presents a variety of diversification strategies to deal with as many image conditions as possible. The selective search algorithm is subject to the following design considerations: The selective search algorithm captures objects at several scales [20], by making hierarchical grouping [21] on the basis of the algorithm; selective search successfully deals with the problem of objects occurring at different levels. There is no single optimal strategy for merging regions together. Regions may form an object because of only color, only texture, or because parts are enclosed. Regions are merged based on different similarity parameters between them, namely: similarity in texture, similarity in color, similarity in fill, similarity in size. The aim of the algorithm points at detecting the object quickly and efficiently without causing a computational bottleneck. Image segmentation is done in order to divide the images based on regions. These regions are then merged using the similarities in texture, color, fill, and size. Selective search takes about 2–8 s for an image to give region proposals.

5 Object Recognition

5.1 Unsupervised Pre-training

Ongoing researches [22] deploy learning algorithms like restricted Boltzmann machines, auto-encoders, and deep belief networks for deep architectures, and good results are obtained. Deep networks work best with an unsupervised component in them.

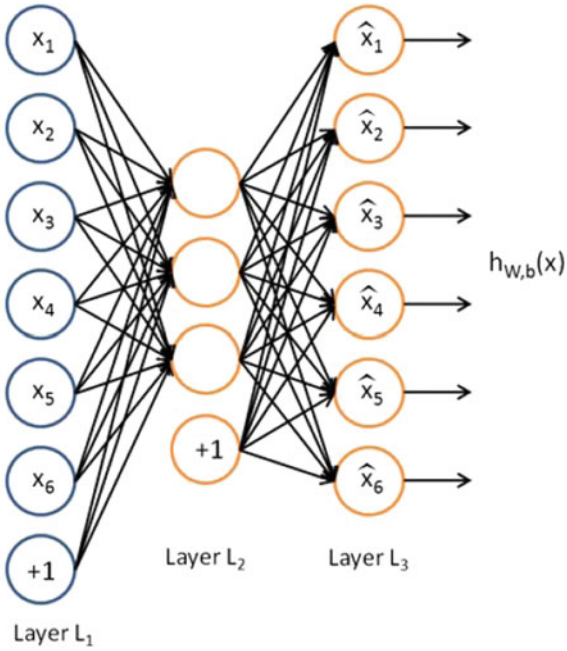


Fig. 1 Sparse auto-encoder [25, 26]

Various studies [22–24] have proven unsupervised pre-training to be the best option to direct the learning convergence toward the minima which is the best-obtained data generalization. This allows us to train better models using a small labeled dataset of images at our disposal.

Unsupervised learning helps in the initialization of weights in the subsequent training procedures. We use a sparse auto-encoder for pre-training in this model.

Preprocessing of Data: The data includes a randomly sampled set of 100,000 patches of size $8 \times 8 \times 3$ from the STL-10 dataset. These patches are subtracted by their mean to get the mean subtracted patches. These patches are flattened to give a linear array of size 192×1. These patches are subsequently zero component analysis (ZCA) whitened. These preprocessing procedures have shown to produce better features by reducing the possibility of 'dead' neurons in the architecture.

Sparse Auto-encoder [25]: The preprocessed data is then fed into a sparse auto-encoder with 192 visible units and 400 hidden units. The final pre-trained features include 400 filters with dimension 192×1. The sigmoidal activation function is used for auto-encoder training which is a nonlinear activation function and gives output between 0 and 1 (refer Fig. 1).

Cost: The cost has three components: the difference in the reproduced image and original, the regularization cost, and the sparsity cost. The sparsity cost uses the KL divergence of the activations of the hidden layer.

Optimization: We use a gradient descent optimizer with an adaptive learning rate [27] to minimize the cost and update weights accordingly. We run the model until a

convergence point is reached in the values of the cost. We saved the learned features, the mean matrix, and the ZCA matrix for the subsequent supervised learning.

5.2 Supervised Convolutional Neural Net

The architecture of the model is largely based on the model described in [28] and [29]. We have a convolutional layer of depth 400, and we use patch sizes of size $8 \times 8 \times 3$.

Convolutional Layer: The convolutional layers use a filter of size $8 \times 8 \times 3$ and depth 400 which gives a feature map of 400 layers. The initial values of weights for this layer are the ones learned during the pre-training stage.

ReLU Layer: In a standard neural layer, the output of a neuron is modeled as an output function y as in terms of its input x is with $y(x) = \tanh(x)$ rather than the sigmoid version $y(x) = \left(1 + e^{-x}\right)^{-1}$. These activation functions are much slower than the non-saturating nonlinearity $y(x) = \max(0,\ x)$. The neurons with this non-linearity are referred to as rectified linear units or ReLUs. We add a ReLU layer after each convolutional layer.

Pooling Layer: We follow the max-pooling algorithm. Small rectangular blocks are taken from the convolutional layer and are subsampled by the pooling layer to produce a single output. The max-pooling layers take the maximum of the block they are pooling. The size of the blocks used in this paper is of size 2×2. Thus, the size of the images is reduced by two every layer.

Fully Connected Layers: We add two fully connected layers for classifying images into the specified classes based on the trained features. These fully connected layers perform a softmax regression to classify the images into the desired number of classes, which are ten in the case of the STL-10 dataset and four in our dataset.

Loss and Optimization: A gradient descent optimizer with an adaptive learning rate with exponential decay is used to minimize the loss. The optimizer is run for 30 epochs for the model to be trained. To reduce the overfitting of features, some measures need to be taken.

Reduction of Overfitting: To reduce overfitting, we regularize the loss with a regularization factor of 0.001. We also add a dropout layer to reduce overfitting [30]. Figure 2 shows the block diagram of proposed network architecture. Figure 3 shows the steps for detection and recognition of objects in a scene image [29].

6 Dataset

The dataset used for testing the algorithm is one standard dataset (STL-10) and a custom four-object dataset. Since the work does is not aiming at improving the accuracy of classification but on localization, the four-object dataset would serve the

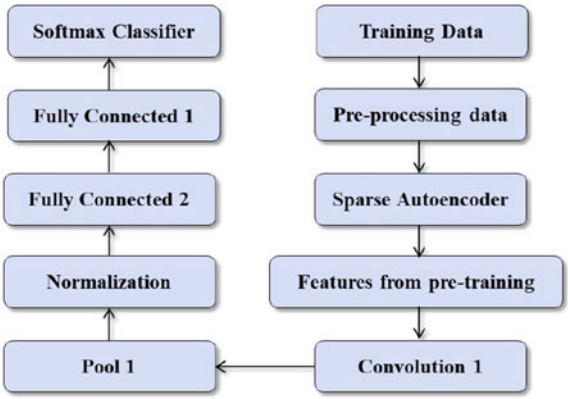


Fig. 2 Proposed network architecture

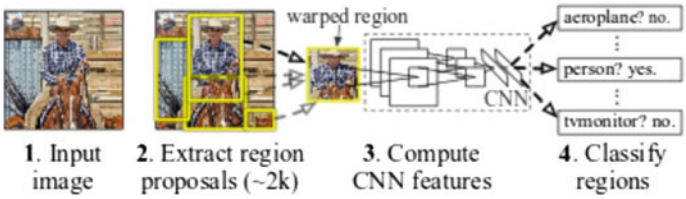


Fig. 3 Detection and recognition of objects in scene [29]

purpose. The model was initially trained and tested on the STL-10 dataset. After the first stage, the training was conducted on a dataset of four objects with 50 labeled images each. In the second stage, 100,000 images from STL-10 database were used for pre-training.

6.1 STL-10 Dataset

The STL-10 dataset consists of 100,000 unclassified images of size $96 \times 96 \times 3$ of objects in 10 classes. These are used to perform unsupervised training. Apart from these, the dataset includes 5000 labeled images of size $96 \times 96 \times 3$ to train the network. The STL-10 dataset was chosen to test upon as the dataset provides a large number of unclassified images and a relatively small number of labeled images to train and test upon. Figure 4 shows the STL-10 dataset.

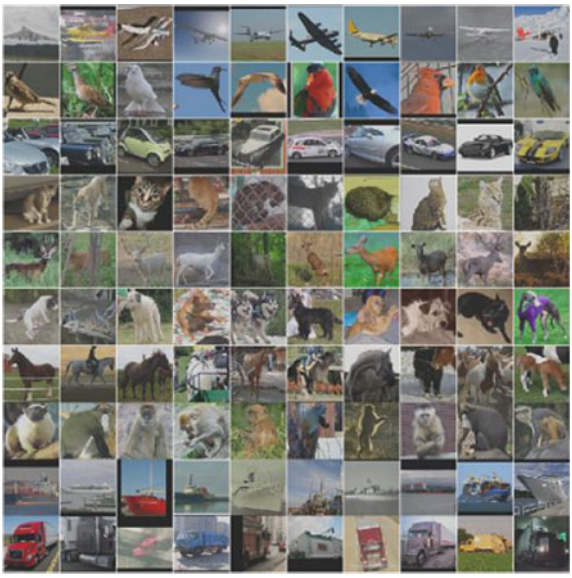


Fig. 4 Ten classes in the STL-10 dataset

6.2 Custom Four-Object Dataset

This dataset includes 4 classes with 50 labeled images of each class, namely: black box, white-yellow box, cold drink paper cup, mosquito repellent. This dataset includes images of these objects from several angles in various lighting conditions with a variable background. The dataset was built in June 2016. It was necessary to build this dataset to test the proposed model on a small, completely new labeled dataset. These images are of high quality of size $3120 \times 3120 \times 3$. The pre-training for these features is done using the STL-10 dataset. The images are centrally cropped and reduced to a size of $64 \times 64 \times 3$ before training the neural network. This dataset was built to test the desired model on a limited number of high-quality labeled images. Figure 5 shows the custom four-object dataset.

7 Results and Discussion

The entire algorithm is tested on the datasets described in Sect. 6 using the algorithms 1 and 2. The entire algorithm is divided into two regions, viz., training and testing which very well explain the process of how the CNN is trained using a small labeled dataset using 100,000 patches for auto-encoder from 15 images. The testing phase best explains how the system is capable of localizing the object using filtering

Fig. 5 Four classes in our dataset

Algorithm 1: Pre-training the CNN

Step 1: Preprocessing of data to remove dead neurons in the architecture.

- The data set images are broken down into random samples of 100,000 patches of size 8x8x3.
- Mean subtracted patches are obtained for normalization.
- Linear array is generated by flattening them to 192x1 linear array.
- ZCA whitening is done.

Step 2: Sparse Auto encoder to learn the features.

- The unit consisting of 192x400x192 neurons.
- The generated 192x1 matrix is the feature matrix associated with the features that are to be scanned.

Step 3: Cost function Optimization.

- The cost function has got the part for difference between the true value and obtained value, sparsity parameter and a regularization factor.
- Gradient descent algorithm is used to optimize the cost by generating updates to make the cost function value reach an optimum.
- The learned features, the mean matrix and the ZCA matrix is passed over for the subsequent analysis.

of segmented regions to propose the best-suited regions to the CNN to scan the area using the selective search algorithm. The Felzenszwalb algorithm and the selective search algorithm were used successfully to segment and filter the regions associated with probable object discovery as explained in Sects. 3 and 4. The regions proposed after the selective searching are forwarded to a pre-trained CNN. The algorithm in the paper thus suggests an efficient way to localize objects by using graph-based segmentation and then merging the desired regions to produce region proposals with a high likelihood of containing an object. These are then fed to a neural network that classifies the object.

The algorithm proved successful in using a large number of sampled patches, to train an auto-encoder in an unsupervised fashion. This pre-training is what is important to train a neural network with a low number of labeled images. The filters learned in the pre-training phase are used to initialize the weights in the supervised learning phase. The success of the final classification governs the accuracy of total algorithm. The supervised learning phase uses a convolutional neural network with a softmax classifier. This network is trained using the small number of labeled images. The testing phase employs the convolutional neural network with region proposals from the selective search algorithm. Once these proposals go through the convolutional neural network, we get a prediction matrix of softmax probabilities for the region describing which class it could belong to.

We choose the region with the maximum probability of containing the object and display a bounding box around it. Pre-training allows us to train a CNN for object recognition using a small amount of labeled data. An object is detected in a given image using a threshold value for the recognition of object. A patch set of 100,000

Algorithm 2: Object Localization Testing

Step 1: Image segmentation and region proposal.

- Uses the graph based Falzenwalb algorithm to segment the image into parts.
- Boundaries determined by graph based image representation.
- Algorithm makes greedy decision to produce segments that satisfy global properties.
- Selective search algorithm merges these segmented regions based on color, texture, size and fill.

Step 2: Feature extraction

- The region proposals are forwarded to the pre-trained CNN having 8x8x3 sized 400 filters to convolve over producing 400 outputs.

Step 3: Classification

- The 400 features generated are then passed to the fully connected layer to classify them to the desired classes.
- The highest value of probability associated will give the best suited class of detection.
- The prediction values associated with each proposed regions are analyzed and the best detection result is taken.

images were given for pre-training, and a set of 10–15 images were given for fine tuning for the classification part. We obtain an accuracy of 85% on the test images. The architecture when tested on the STL-10 dataset gave a state-of-the-art accuracy of 80%.

The regions proposed and the final bounding box based on the predictions from the CNN are generally accurate, but at times the bounding box is larger than required while sometimes it turns out to merely include a part of the object. The results of segmentation and selective searching tested on the standard test images are as shown in Figs. 6 and 7.

Fig. 6 Detection and recognition of objects in scene 1: gray scale [18]

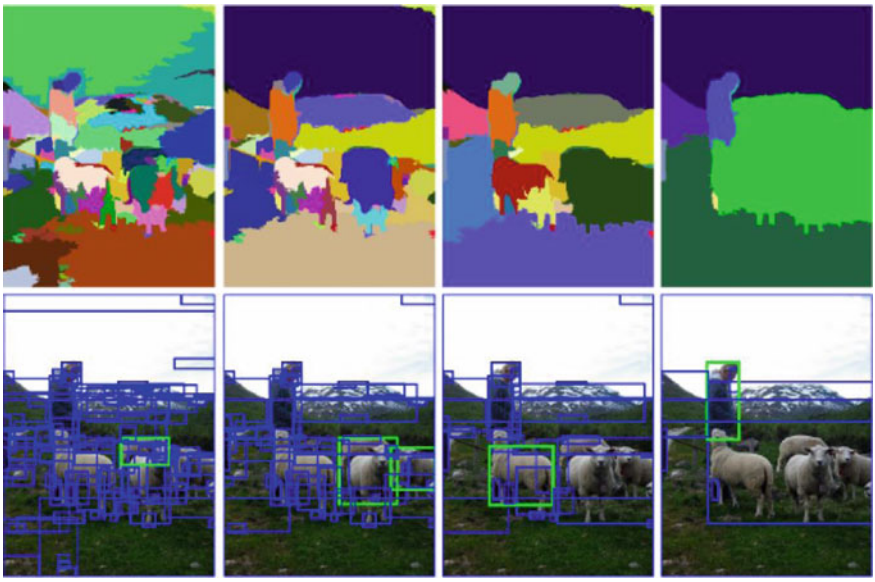


Fig. 7 Detection and recognition of objects in scene 2: RGB [18]

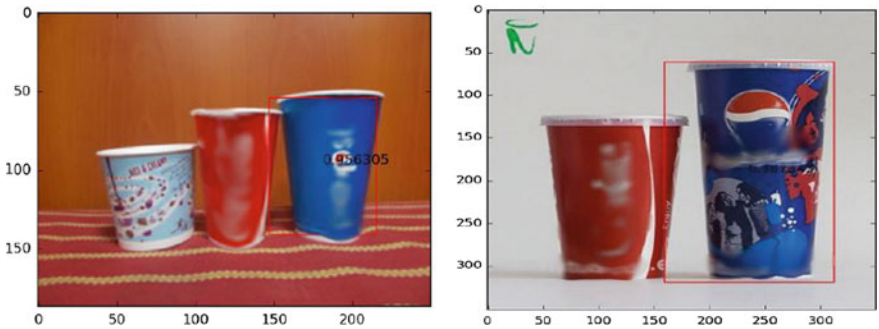


Fig. 8 Detection and recognition of glass

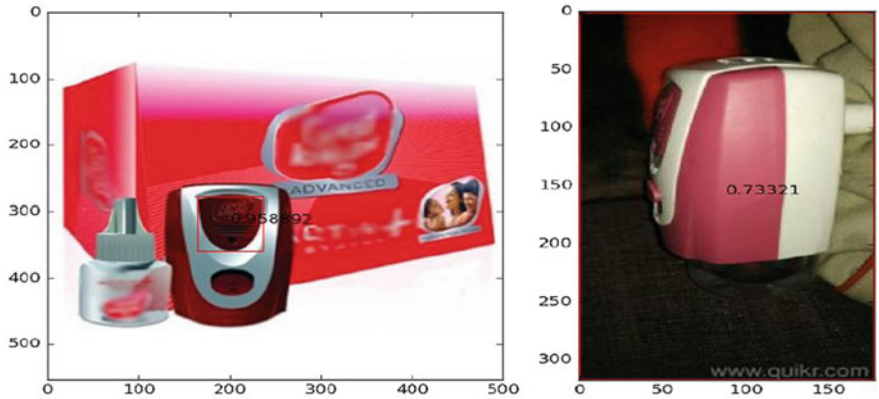


Fig. 9 Detection and recognition of mosquito repellent

The localization can be merely represented by the coordinates of a bounding box. The experiment conducted with custom four-object class dataset has shown good detection result with accuracy around 85%. The results of detection are as shown in Figs. 8 and 9.

8 Conclusions

This paper presents a methodology to detect and recognize objects in an image. This object segmentation/detection algorithm was developed mainly for datasets with a lower number of labeled images. The main idea behind the algorithm is to provide region proposals to a deep network to classify the images and provide a bounding box around it. The problem that arises a great number of times is to find a large dataset to train the network. This problem is overcome to a great extent by a good

pre-training algorithm. This paper proposes one such architecture that can be used for object detection.

8.1 Limitations and Future Work

The accuracy can probably be improved by varying the parameters and architecture of the model proposed.

(1) The region proposal algorithm of selective search takes up a lot of time proposing the regions making the process extremely expensive. Exploring other region proposal algorithms could certainly be used to make the process quicker.
(2) The accuracy of the recognition model can be improved by using more complex pre-training methods like deep belief networks. A larger amount of labeled data can also be used to improve accuracy. This paper measures and compares the performance of some commonly used pre-training methods.
(3) The parameters of the existing architecture can be worked upon to get the most optimal results in the minimal amount of data.

References

1. Pinto, N., Cox, D.D., DiCarlo, J.J.: Why is real-world visual object recognition hard? PLoS Comput. Biol. **4**(1), e27 (2008)
2. Verma, N.K., Sahu, S.K., Mustafa, A., Dhar, N.K., Salour, A.: Priority based optimal path routing for automated guided vehicle. In: IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions, pp. 1–7 (2015)
3. Verma, N.K., Sunny, N.K., Mishra, A.: Generation of future image frame using autoregressive model. In: IEEE Conference on Industrial Electronics and Applications, pp. 171–176. Auckland, New Zealand (2015)
4. Verma., N.K., Mishra, A.: Large displacement optical flow based image predictor model. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, Oct 2014
5. Verma, N.K., Singh, S.: Generation of future image frames using optical flow. In: Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–7. Washington DC, USA, 23–25 Oct 2013
6. Agrawal, P., Sharma, T., Verma, N.K.: Supervised approach for object identification using speeded up robust features. Int. J. Adv. Intell. Paradigms (IJAIP) (2017) (Accepted for publication)
7. Verma, N.K., Sharma, T., Sevakula, R.K., Salour, A.: Vision based object counting using speeded up robust features for inventory control. In: International Conference on Computational Science and Computational Intelligence (CSCI'16). Las Vegas, Nevada, USA, 15–17 Dec (2016) (In Proceedings)
8. Verma, N.K., Sharma, T., Rajurkar, S.D., Salour, A.: Object identification for inventory management using convolutional neural network. In: IEEE Applied Imagery Pattern Recognition Workshop (AIPR). Washington DC, USA, 18–20 Oct (2016) (In Proceedings)

9. Verma, N.K., Sharma, T., Rajurkar, S.D., Ranjan, R., Salour, A.: Vision based counting of texture-less objects using shape and color features. In: IEEE International Conference on Industrial and Information Systems (ICIIS). IIT Roorkee, India, 3–4 Dec (2016) (In Proceedings)
10. Verma, N.K., Sharma, T., Rajurkar, S.D., Molangur, N., Salour, A.: Multi-faced object recognition in an image for inventory counting. In: IEEE International Conference on Design and Management (IConDM). IIITDM Kancheepuram, Chennai, India, 16–17 Dec (2016) (In Proceedings)
11. Verma, N.K., Kumar, G., Siddhant, A., Nama, P., Raj, A., Mustafa, A., Dhar, N.K., Salour, A.: Vision based obstacle avoidance and recognition system. In: IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions, pp. 1–7 (2015)
12. Verma, N.K., Nama, P., Kumar, G., Siddhant, A., Raj, A., Dhar, N.K., Salour, A., et al.: Vision based object follower automated guided vehicle using compressive tracking and stereo-vision. In: IEEE Bombay Section Symposium (IBSS), pp. 1–6 (2015)
13. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II-53 (2004)
14. Swain, M.J., Ballard, D.H.: Color indexing. Int. J. Comput. Vis. **7**(1), 11–32 (1991)
15. Porikli, F.: Integral histogram: a fast way to extract histograms in cartesian spaces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 829–836 (2005)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
17. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vis. **59**(2), 167–181 (2004)
18. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
19. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2189–2202 (2012)
20. Arbelaez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 328–335 (2014)
21. Arbelaez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., Malik, J.: Semantic segmentation using regions and parts. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3378–3385 (2012)
22. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. **11**(Feb), 625–660 (2010)
23. Erhan, D., Manzagol, P.-A., Bengio, Y., Bengio, S., Vincent, P.: The difficulty of training deep architectures and the effect of unsupervised pre-training. In: AISTATS, vol. 5, pp. 153–160 (2009)
24. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. Adv. Neural. Inf. Process. Syst. **19**, 153 (2007)
25. Ng, A.: Sparse autoencoder. CS294A Lecture Notes, vol. 72, pp. 1–19 (2011)
26. Autoencoders and sparsity. In Autoencoders and Sparsity. http://udl.stanford.edu
27. Jacobs, R.A.: Increased rates of convergence through learning rate adaptation. Neural Netw. **1**(4), 295–307 (1988)
28. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. proc. Syst. 1097–1105 (2012)
29. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
30. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

Part VI
Optimization Algorithms

Multi-objective OPF Problem Analysis with Practical Constraints in the Presence of FACTS Devices Using NSHCSA

M. Balasubbareddy

Abstract This research paper proposes a hybrid cuckoo search algorithm (HCSA) for OPF problem solution in power systems. In this, genetic algorithm (GA) is combined with conventional cuckoo search algorithm (CSA) to improve the performance of the single-objective and multi-objective problem solution with satisfying equality, inequality, and practical constraints such as ramp-rate limits and prohibited operating zones (POZ). Fuzzy approach is used to select optimal solution required by the user from the total solutions. The proposed NSHCSA with TCSC, SSSC, and IPFC is tested on standard test function and IEEE 14 bus system, and corresponding results are analyzed.

Keywords Hybrid cuckoo search algorithm · Multi-objective optimization Non-dominated sorting · TCSC · SSSC · IPFC

1 Introduction

For operation and planning of modern power systems, the optimal power flow is used. Sun et al. [1] have given Newton method to solve the classical optimal power flow problem with a non-separable objective function. Yuryevich et al. [2] proposed new evolutionary programming algorithm for OPF problem solution to improve the speed of convergence of the algorithm as well as its ability to handle larger systems. Yan and Quintana [3] proposed interior point algorithm for solving optimal power flow problems. da Costa and Costa [4] proposed a novel algorithm to improve the performance of the classical methods for OPF problem. Fuerte-Esquivel et al. [5] proposed firing angle model of thyristor-controlled series compensator (TCSC).

M. Balasubbareddy (✉)
Chaitanya Bharathi Institute of Technology (Autonomous), Hyderabad, India
e-mail: balasubbareddy79@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_32

From the standard existing research data, it is observed that OPF problem is solved without considering the practical constraints, location, and cost of FACTS devices. In this paper, we consider the above drawbacks, and the results obtained are better than the existing methods.

2 Modeling of FACTS Devices

A mathematical model based on power injections is used to study the performance of the FACTS devices on power system problems, and this model is added in the load flow method for the analysis. The power injection model of TCSC, SSSC, and IPFC is derived in [6–8], respectively.

3 Cost of FACTS Devices

In this research, we consider the cost of the FACTS devices is incorporated along with the cost objective function. In this, 15-year duration is considered for analysis and installation cost of TCSC, SSSC, and IPFC is derived in [9–11].

4 Problem Formulation

The mathematical equation for the objective function is

$$Min[F_T(x,\ u)]$$
$$\text{Subjected to} \begin{matrix} m(x,\ u) = 0 \\ n(x,\ u) \leq 0 \end{matrix} \tag{1}$$

4.1 Objective Functions

(i) **Generation Cost Minimization**
The generation cost is $F_i(P_{Gi}) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i\ \$/h$

$$Total\ cost\ A_1 = \min(F_{P_{Gi}}) + C_{FACTS}\ \$/\text{h} \tag{2}$$

(ii) **Power Loss Minimization**

$$A_2 = \min\left(\sum_{i=1}^{N_{line}} P_{Loss,i}\right) \tag{3}$$

4.2 Constraints

(i) Equality constraints

$$\sum_{k=1}^{n_G} P_{G_k} - PD - PL = 0$$
$$\sum_{k=1}^{N_G} Q_{G_k} - QD - QL = 0$$

(ii) Inequality constraints

$$V_{G_k}^{\min} \le V_{G_k} \le V_{G_k}^{\max} \quad \forall k \in N_G$$
$$P_{Gk}^{\min} \le P_{G_k} \le P_{G_k}^{\max} \quad \forall k \in N_G$$
$$T_k^{\min} \le T_k \le T_k^{\max} \quad k = 1,\ 2,\ \ldots,\ n_t$$
$$Q_{C_i}^{\min} \le Q_{C_i} \le Q_{C_i}^{\max} \quad i = 1,\ 2,\ \ldots,\ n_C$$
$$S_{l_k} \le S_{l_k}^{\max} \quad k = 1,\ 2,\ \ldots,\ N_{line}$$
$$Q_{G_k}^{\min} \le Q_{G_k} \le Q_{G_k}^{\max} \quad \forall k \in N_G$$
$$V_k^{\min} \le V_k \le V_k^{\max} \quad k = 1,\ 2,\ \ldots,\ N_{load}$$

(iii) TCSC constraint

$$-0.8\, X_{line} \le X_{TCSC} \le 0.2 X_{line}\ p.u.$$

(iv) SSSC constraints

$$0 \le V_{se} \le 0.1\ p.u.; 0 \le \theta_{se} \le 360^{\circ};\ 0 \le X_{se} \le 0.1\ p.u.$$

(v) IPFC limits

$$0 \le V_{se_A} \le 0.1\ p.u.,\ 0 \le V_{se_B} \le 0.1\ p.u.,\ 0 \le \theta_{se_A} \le 360^{\circ}$$
$$0 \le \theta_{se_B} \le 360^{\circ},\ 0 \le X_{se_B} \le 0.1\ p.u.,\ 0 \le X_{se_B} \le 0.1\ p.u.$$

(vi) Prohibited operating zones (POZs) (practical constraints)
The POZ can be included as follows:

$$P_k = \begin{cases} P_k^{\min} \le P_k \le P_{k,1}^{L} \\ P_{k,j-1}^{U} \le P_k \le P_{k,j}^{L} \quad j = 2,\ 3,\ \ldots,\ n_k \\ P_{k,n_k}^{U} \le P_k \le P_k^{\max} \end{cases}$$

(vii) Ramp-rate limits

$$\max\left(P_{Gk}^{\min},\ P_k^0 - DR_k\right) \le P_{Gk} \le \min\left(P_{Gk}^{\max},\ P_k^0 + UR_k\right)$$

here P_k^0 is the generation of kth unit at earlier hour. DR_k are the respective increasing and decreasing ramp-rate limits of ith unit.

Equation (4) can be written as

$$A_{m,aug}(x,\ u) = A_m(x,\ u) + K_1\left(P_{g,slack} - P_{g,slack}^{\lim}\right)^2 + K_2\sum_{i=1}^{N_{Load}}\left(V_i - V_i^{\lim}\right)^2 + K_3\sum_{i=1}^{N_G}\left(Q_{G_i} - Q_{G_i}^{\lim}\right)^2 + K_4\sum_{i=1}^{N_{line}}\left(S_{l_i} - S_{l_i}^{\max}\right)^2 \tag{4}$$

where $K_1,\ K_2,\ K_3$, and K_4 are the penalty factors considered high positive values. The limits are as follows:

$$x^{\lim} = \begin{cases} x^{\max},\ x > x^{\max} \\ x^{\min},\ x < x^{\min} \end{cases}$$

5 Optimal Location of FACTS Devices

In this paper, selection of the location is based on the minimization of the severity function value. The function is given as follows:

$$F_{Severity} = \sum_{k=1}^{N_{line}}\left(\frac{S_k}{S_k^{\max}}\right)^{2q} + \sum_{i=1}^{N_{bus}}\left(\frac{V_{i,ref} - V_i}{V_{i,ref}}\right)^{2r} \tag{5}$$

6 Optimal Location of FACTS Devices

Algorithm steps are as follows:

(i) Initialization
Random generation is given by

$$x_{pq} = x_q^{\min} + rand(0,\ 1) \times \left(x_q^{\max} - x_q^{\min}\right)$$

(ii) Levy flights
To lay egg, cuckoo chose nest position randomly, and levy flight expression is used to generate new solution [12].

$$x_i(t+1) = x_i(t) + S_{pq} \times \alpha \ \oplus \ Levy(\lambda) \tag{6}$$

where $\alpha = \left[-1\ 1\right]$
$\oplus =$entry-wise multiplication

$$S_{pq} = x_{pq}^t - x_{fq}^t;$$

where $p,\ f\ = 1,\ 2,\ \ldots,\ n$ and $q = 1,\ 2,\ 3,\ \ldots,\ m$

$$Levy(\lambda) = \left| \frac{\Gamma(1+\lambda) \times \ \sin\left(\frac{\pi \times \lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right) \times \lambda \times 2^{\left(\frac{\lambda-1}{2}\right)}} \right|^{1/\lambda}; \quad 1 < \lambda \leq 3 \tag{7}$$

(iii) Cross over
GA-based crossover operations are used for search process [13]

$$x_{pq}^{new} = (1-\lambda) \times x_{1q}^{ref} + \lambda \times x_{pq}^{old} \tag{8}$$

where $\lambda = \left[0\ 1\right]$

(iv) Selection
Based on the fitness values, sorting and ranking processes take place. The best population values are used for next iteration till the maximum iterations reached.

(v) Stopping criteria
Stop iteration after reaching the maximum specified number of iterations.

7 Optimal Location of FACTS Devices

As per the user, select the best solution from the multiple solutions obtained from multi-objective optimization problem. For this process, fuzzy decision tool is used. The membership function is given by

$$\mu_i^j = \begin{cases} 1; & A_i^j \leq \min(A_i) \\ \frac{\max(A_i)-A_i^j}{\max(A_i)-\min(A_i)}; & \min(A_i) \leq A_i^j \leq \max(A_i) \\ 0; & A_i^j \geq \max(A_i) \end{cases} \tag{9}$$

$$\mu_{opt} = \underbrace{\sup}_{n \in j} \frac{\sum_{i=1}^{p} W_i \mu_m^n}{\sum_{n=1}^{M} \sum_{i=1}^{p} W_i \mu_m^n}; \quad \text{Where } W_i \geq 0; \quad \sum_{i=1}^{p} W_i = 1 \tag{10}$$

8 Multi-objective Optimization

The complete NS and distance calculations are described in [14, 15]. The proposed multi-objective optimization process is described as follows:

Step-1 Generate the initial search agents.
Step-2 Perform Newton–Raphson load flow with FACTS device.
Step-3 Treat this solutions as parent solutions.
Step-4 Find its dominated agents and find the different Pareto fronts ***Pf***.
Step-5 Find the individual solution in the next non-dominated front.
Step-6 Perform NS on total solution.
Step-7 Crowding distance operation is performed.
Step-8 For each objective solution, assign boundary values as infinite distance.
Step-9 Perform this for all the objectives.
Step-10 As per the crowing distance, sorting is performed and increases the iteration count.
Step-11 Repeat the steps 5–8 till maximum iterations reached.
Step-12 Calculate Pareto fronts using fuzzy membership tool.

9 Results and Analysis

In this paper, two examples are considered: Himmelblau function and IEEE 14 bus systems with different FACTS devices to validate the effectiveness of the proposed technique.

9.1 Example-I

To test the efficacy of the proposed technique, a standard Himmelblau function [13] is to be minimized in the limits ($0 \leq x_1,\ x_2 \leq 6$) and is as follows:

$$f(x_1,\ x_2) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2 \tag{11}$$

Results obtained for optimal solution with 100 iterations from PSO, CSA, and HCSA including GA method are compared and given in Table 1. Convergence characteristics obtained with PSO, CSA, and HCSA methods are shown in Fig. 1.

Table 1 OPF solution obtained for test function

S. No.	Parameters	PSO	CSA	Proposed HCSA
1	x_1	3.0174	2.9902	3.0013
2	x_2	1.9681	2.0058	1.9993
3	Function value	0.0172	0.0029	4.95644E–05
4	Computational time (s)	1.657	0.865	0.521

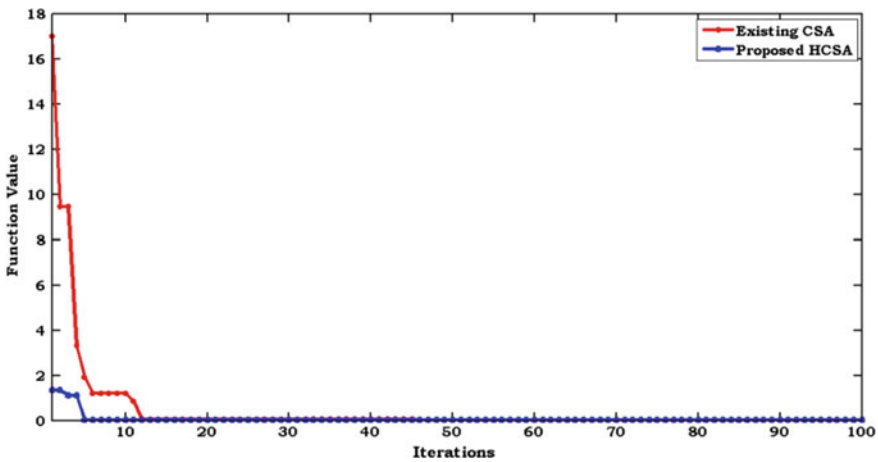


Fig. 1 Comparison of convergence characteristics of Himmelblau function

From Table 1 and Fig. 1, it is observed that the results obtained from PSO, CSA, and proposed HCSA methods are better than existed GA method. Finally, it is concluded that results obtained with proposed HCSA method are better and optimum manner than the existing PSO and CSA.

9.2 Example-II

An IEEE 14 bus system [16] is considered for validation of the proposed method.

Optimal Location: As per the contingency analysis and considered heuristic rules, the contingency ranking is given in Table 2. From Tables 2, 3, and 4, the TCSC and SSSC are located in 12 and 13 buses. And IPFC is placed between buses 13–12 and 13–14.

Single-objective Analysis: The single-objective results are given in Table 5. From Table 5, the observations are as follows:

- The cost of generation and transmission losses is less with IPFC when compared to without FACTS, with TCSC, and with SSSC. With IPFC, net saving is 2.83 $/h.
- From the analysis, it is observed that system security is improved by SSSC device when compared to other devices described above.

Table 2 Contingency analysis ranking

S. No.	Outage line	NOLL	NVVB	PI	Rank
01	1–5	04	0	04	03
02	2–3	04	0	04	04
03	2–4	03	0	03	09
04	2–5	03	0	03	10
05	3–4	04	0	04	05
06	**4–5**	**07**	**0**	**07**	**01**

Table 3 Best possible locations in IEEE 14 bus system of TCSC and SSSC with severity

S. No.	Sending-end bus	Receiving-end bus	Severity	
			TCSC	SSSC
1	10	11	2.5981	2.4432
2	**12**	**13**	**2.5744**	**2.4321**
3	13	14	2.5794	2.4783

Table 4 Best location of IPFC

Location	SE bus	RE bus-1	RE bus-2	Value of severity
1	13	12	14	2.31

Table 5 Summary of single-objective optimization results

Control variables	Generation fuel cost ($/h)				Total power loss (MW)			
	Without FACTS	With TCSC	With SSSC	With IPFC	Without FACTS	With TCSC	With SSSC	With IPFC
V_{se1} (pu)			0.00001	1.0e–6			0.001	0.001
V_{se2} (pu)				0.014				0.035
θ_{se1} (°)			360	110.482			286.167	360
θ_{se2}(°)				360				279.79
X_{se1} (pu)			0.040	0.092			0.05	0.1
X_{se2} (pu)				0.078				0.093
Cost ($/h)	721.833	720.995	720.324	719.00	862.82	1103.05	980.47	1087.21
P_{Loss} (MW)	8.314	9.807	8.268	8.555	4.88	4.77	4.307	4.12

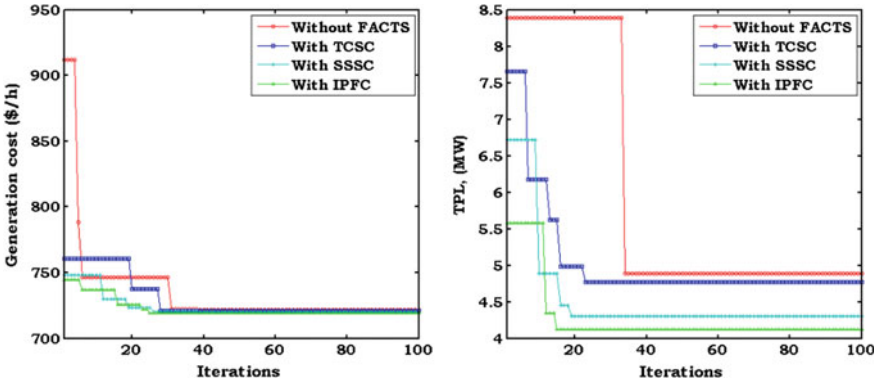


Fig. 2 Convergence characteristics of fuel cost and transmission power loss for IEEE 14 bus system with and without FACTS devices

The convergence characteristics with different FACTS devices are shown in Fig. 2, and from these, it is observed that for all considered objective functions with IPFC converge the solution with less time and number of iterations.

Multi-objective Optimization Result Analysis: In this analysis, objective functions are solved using proposed NSHCSA for considered objectives with and without FACTS devices.

From Fig. 3, it shows the evidence that with IPFC gives the best Pareto fronts when compared to without and with TCSC and SSSC devices. The chosen Pareto fronts for cost-loss combination of two objectives with different weights are given in Table 6. It is observed that based on the weights given to the objective function, the objective value varies accordingly. If we give more weight age to the particular function, that function will minimize more compared to other less weight age functions.

Table 6 Multi-objective results of IEEE 14 bus system

S. No.	W_1	W_2	Without FACTS		With TCSC		With SSSC		With IPFC	
			Cost ($/h)	Loss (MW)	Cost ($/h)	Loss (MW)	Cost ($/h)	Loss (MW)	Cost ($/h)	loss (MW)
1	0.9	0.1	742.293	8.632	738.756	8.295	732.638	9.446	729.925	9.198
5	0.5	0.5	818.886	6.980	794.080	6.721	766.044	7.131	745.402	7.919
9	0.1	0.9	1051.474	6.077	901.068	5.807	909.191	5.655	995.891	5.573

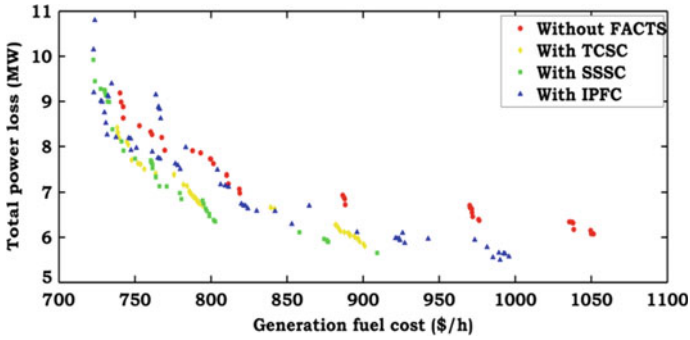


Fig. 3 Best Pareto front without FACTS and with FACTS devices

10 Conclusions

In this paper, the proposed NSHCSA algorithm efficiently optimized the single object such as generation cost, power loss, and multi-objective functions with satisfying operating and practical constraints. Location of FACTS device is determined based on the enhancement of system security. The proposed algorithm was tested with two examples to find the optimal control variables. The results show the best when compared to the existing literature.

References

1. Sun, D.I., Ashley, B., Brewer, B., Hughes, A.: Optimal power flow by Newton approach. IEEE Trans. Power Appar. Syst. **PAS-103**(10) (1984)
2. Yuryevic, J., Wong, K.P.: Evolutionary programming based optimal power flow algorithm. IEEE Trans. Power Syst. **14**(4) (1999)
3. Yan, X., Quintana, V.H.: Improving an interior-point-based OPF by dynamic adjustments of step sizes and tolerances. IEEE Trans. Power Syst. **14**(2) (1999)
4. da Costa, G.R.M., Costa, C.E.U.: Improved newton method for optimal power flow problem. Electr. Power Energy Syst. **22**, 459–462 (2000)
5. Fuerte-Esquivel, C.R., Acha, E., Ambriz-PBrez, H.: A thyristor controlled series compensator model for the power flow solution of practical power networks. IEEE Trans. Power Syst. **15**(1) (2000)
6. Ongsakul, W., Bhasaputra, P.: Optimal power flow with FACTS devices by TS/SA approach. Electr. Power Energy Syst. **24**, 851–857 (2002)
7. Zhang, X.-P.: Advanced modeling of the multicontrol functional static synchronous series compensator (SSSC) in Newton power flow. IEEE Trans. Power Syst. **18**(4), 1410–1416 (2003)
8. Zhang, X.P.: Modeling of the interline power flow controller and the generalized unified power flow controller in Newton power flow. IEEE Proc. Gener. Transm. Distrib. **150**(3), 268–274 (2003)
9. Xiao, Y., Song, Y.H., Sun, Y.Z.: Power flow control approach to power systems with embedded FACTS devices. IEEE Trans. Power Syst. **17**(4), 943–950 (2002)
10. Nabavi, S.M.H., Khafafi, K., Sakhavati, A., Nahi, S.: Optimal location and sizing of SSSC using genetic algorithm in deregulated power market. Int. J. Comput. Appl. **22**(4), 37–41

11. Sreejith, S., Psimmon, Sishaj, Selvan, M.P.: Optimal location of interline power flow controller in a power system network using ABC algorithm. Archives of Electr. Eng. **62**(1), 91–110 (2013)
12. Yang, X.S., Deb, S.: Cuckoo search via levy flights. In: Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), India, pp. 210–214. IEEE Publications, USA
13. Deb, K.: Optimization for Engineering Design: Algorithms and Examples, pp. 303–316. PHI (2009)
14. Alsac, O., Stott, B.: Optimal load flow with steady state security. In: IEEE PES Summer Meeting & EHV/UHV Conference, July, 1973, pp. 745–751
15. Srinivas, N., Deb, K.: Multi objective function optimization using non dominated sorting genetic algorithms. Evol. Comput. **2**(3), 221–248 (1994)
16. Deb, K., Agarwal, S., Pratap, A., Meyarivan, T.: A Fast elitist non-dominated sorting genetic algorithm for multi-objective optimization. In: NSGA-II, Kanpur Genetic Algorithms Laboratory, IIT Kanpur

A Unified TOPSIS Approach to MADM Problems in Interval-Valued Intuitionistic Fuzzy Environment

Samir Kumar and Animesh Biswas

Abstract In this paper, TOPSIS is extended in interval-valued intuitionistic fuzzy settings using mathematical programming and normalized hamming distance between two intuitionistic fuzzy sets with three parameter characterizations. The two formulae giving lower and upper bounds of intervals for relative closeness coefficients of alternatives are generated on solving two linear fractional programming models. A fuzzy preference relation as possibility degree matrix is formed using pairwise comparisons of intervals of closeness coefficients and used to estimate the optimal degrees for ultimate ordering of alternatives. A numerical example with two distinct forms demonstrates the validity and effectiveness of the presented approach.

Keywords Intuitionistic fuzzy sets · Interval-valued intuitionistic fuzzy numbers TOPSIS · Possibility degree matrix · Mathematical programming

1 Introduction

The technique for order preference by similarity to ideal solution (TOPSIS) advanced by Hwang and Yoon [15] is a highly acclaimed multiple attribute decision-making (MADM) method. In TOPSIS, the decision matrix is either in commensurate units or else normalized using an appropriate normalization procedure. The positive ideal solution (PIS) is generated as a combination of the greatest and the least performance values of each alternative relative to each benefit and cost criteria, respectively, as represented by the decision matrix. Similarly, the negative ideal solution (NIS) is formed as a combination of the least and the greatest performance values of each

S. Kumar
Department of Mathematics, Acharya Jagadish Chandra Bose College, Kolkata 700020, India
e-mail: kumarsamir2007@gmail.com

A. Biswas (✉)
Department of Mathematics, University of Kalyani, Kalyani 741235, India
e-mail: abiswaskln@rediffmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_33

alternative relative to each benefit and cost criteria, respectively, as given by the decision matrix. Thus, the PIS and the NIS are the best and the worst possible alternatives, respectively. The basic principle of the TOPSIS is that the most desirable alternative should be nearest to the PIS and farthest from the NIS. The distance is measured by using appropriate metric in a given decision system. The immense popularity of TOPSIS needed further scientific and mathematical tools and techniques to model and interpret real-life uncertainties [7, 8, 13]. In this quest, Chen [11] utilized the idea of fuzzy sets (FSs) [25] and propounded the fuzzy TOPSIS concept drawing the scrutiny of researchers [1, 19, 20]. In spite of several applications of FSs [5, 9, 10], the concept got generalized to intuitionistic fuzzy sets (IFSs) [2, 4] and further to interval-valued intuitionistic fuzzy sets (IVIFSs) [3]. As a consequence, the fuzzy TOPSIS was also extended to intuitionistic fuzzy (IF) TOPSIS [12, 17, 18, 26, 27] for accurate modeling of reality. Zeng and Xiao [26] proposed an IF hybrid ordered weighted averaging distance TOPSIS procedure for solving MCDM problems in IF ambience. Chen [12] utilized the idea of inclusion comparison possibilities and extended the TOPSIS in interval-valued IF (IVIF) environment by formulating relative closeness coefficient (RCC) for ordering of alternatives. Zhao [27] advanced an optimization model for evaluating attribute weights and developed a TOPSIS approach applying weighted hamming distance in IVIF settings for handling MADM problems. The two programming approaches, nonlinear and linear, based on the weighted Euclidean distance [17] and the weighted absolute distance [18], respectively, between two IFSs relative to intuitionistic fuzzy weights were advanced considering two parameter representation of IFSs. Szmidt and Kacprzyk [21] demonstrated that any generalization of fuzzy distance in the IF settings should be expressed in terms of the three parameters, belongingness, non-belongingness, and hesitancy degrees, and the normalized hamming distance is one of the four fundamental distances between the IFSs proposed by them using three parameter characterizations (TPCs) of IFSs. The works of Li [17, 18] and others employ extensive mathematical computations and two parameter IFS representation in IF distance measure (IFDM) in contravention of observations of Szmidt and Kacprzyk [21]. In this context, Biswas and Kumar [6] conceptualized TOPSIS in IVIF settings by employing TPC of IFS in IFDM and programming skills (PSs) illustrating an MADM problem.

The TPC of IFSs [21, 23] empowers decision makers with greater liberty in comparison to FSs where the decision makers are forced between complementary choices of membership and non-membership degrees with no hesitation for judgments in a given decision context. The points $\langle 1, 0, 0\rangle$, $\langle 0, 1, 0\rangle$, and $\langle 0, 0, 1\rangle$ in three-dimensional interpretation of real-life decision making are identified, respectively, with complete acceptance, total denial and absolute hesitation concerning the opinion of an expert.

In this paper, the previous works [6, 17, 18] are further elucidated by exerting the PSs and IFDM with TPC of IFS extending the TOPSIS in IVIF settings and illustrated an MADM problem considering both IVIFN and crisp number with two distinct forms of criteria, viz., benefit and cost.

The rest of the paper is organized as follows. Section 2 reviews the concepts of IVIFSs and ranking of interval numbers using possibility degree measures.

Section 3 introduces programming-based TOPSIS approach in IVIF settings for solving MADM problems with weights of criteria or ratings of alternatives expressed in TPCs of IVIFSs. The validity and effectiveness of the presented approach are illustrated through a numerical example with two distinct forms of opinions, viz., IVIFNs and crisp and two types of criteria, viz., benefit and cost, in Sect. 4 while concluding remarks are appended in Sect. 5.

2 Preliminaries

Definition 2.1 [3]. An IVIFS, A, on a universe of discourse X is defined as a set $A = \{\langle x,\ \mu_A(x),\ \nu_A(x)\rangle : x \in X\}$ such that $0 \leq sup\mu_A(x) + sup\nu_A(x) \leq 1$, for each $x \in X$, where $\mu_A : X \to S[0,\ 1]$ and $\nu_A : X \to S[0,\ 1]$, respectively, represent membership and non-membership functions and $S[0,\ 1]$ constitutes the set of all subsets of interval [0, 1].

The interval degree of membership and non-membership to the set A for each element $x(\in X)$ are represented, respectively, by $\mu_A(x)$ and $\nu_A(x)$. For simplicity, assume that $\mu_A^l(x),\ \mu_A^u(x)$ and $\nu_A^l(x),\ \nu_A^u(x)$ denote, respectively, the lower and upper bounds of the interval degrees $\mu_A(x)$ and $\nu_A(x)$, for each $x \in X$.

The interval hesitancy degree of x in IVIFS A, characterized by function $\pi_A : X \to S[0, 1]$, is expressed by $\pi_A(x) = \left[\pi_A^l(x),\ \pi_A^u(x)\right] = \left[1 - \mu_A^u(x) - \nu_A^u(x),\ 1 - \mu_A^l(x) - \nu_A^l(x)\right]$, $\forall x \in X$.

Thus, the TPC of IVIFS A can be represented as

$$A = \left\{\left\langle\left[\mu_A^l(x),\ \mu_A^u(x)\right],\ \left[\nu_A^l(x),\ \nu_A^u(x)\right],\ \left[\pi_A^l(x),\ \pi_A^u(x)\right]\right\rangle : x \in X\right\}.$$

For each $x \in X$, Xu and Yager [23] called the ordered triple $\langle\mu_A(x),\ \nu_A(x),\ \pi_A(x)\rangle$ an interval-valued intuitionistic fuzzy number (IVIFN). For conciseness, assume that $A = \langle\mu_A,\ \nu_A,\ \pi_A\rangle$ expresses an IVIFN.

Obviously, $\zeta^+ = \langle[1,\ 1],\ [0,\ 0],\ [0,\ 0]\rangle$ and $\zeta^- = \langle[0,\ 0],\ [1,\ 1],\ [0,\ 0]\rangle$ represent, respectively, the greatest and the smallest IVIFNs.

Definition 2.2 [14]. Assume that $a_i = \left[a_i^l,\ a_i^u\right](i = 1,\ 2)$ be two interval numbers with interval lengths $\mathrm{L}(a_i) = a_i^u - a_i^l$. The possibility degree of $a_1 \geq a_2$ is defined as

$$p(a_1 \geq a_2) = min\left\{max\left\{\frac{a_1^u - a_2^l}{L(a_1) + L(a_2)},\ 0\right\},\ 1\right\}$$

Definition 2.3 [16]. Assume that $\alpha_i = \left\langle\left[a_i^l, a_i^u\right],\ \left[b_i^l, b_i^u\right]\right\rangle = \langle a_i,\ b_i\rangle(i = 1,\ 2)$ be two IVIFNs and $\mathrm{L}(a_i) = a_i^u - a_i^l$, $L(b_i) = b_i^u - b_i^l$ be the lengths of the intervals $a_i,\ b_i$, respectively, then the degree of possibility of $\alpha_1 \geq \alpha_2$ is defined as

$$p(\alpha_1 \geq \alpha_2) = min\left\{max\left\{\frac{a_1^u - a_2^l + b_2^u - b_1^l}{L(a_1) + L(a_2) + L(b_1) + L(b_2)},\ 0\right\},\ 1\right\}$$

3 TOPSIS Approach to MADM in Interval-Valued Intuitionistic Fuzzy Settings with Three Parameter Characterizations of IFS

Let $A = \{\langle \mu_A(x_j),\ \nu_A(x_j),\ \pi_A(x_j)\rangle | x_j \in X\}$ and $B = \left\{\left\langle \mu_B(x_j),\ \nu_B(x_j),\ \pi_B(x_j)\right\rangle | x_j \in X\right\}$ be two IFSs in a universe of discourse $X = \{x_1,\ x_2,\ \ldots,\ x_n\}$. Then, the normalized hamming distance [21] between IFSs A and B is given by

$$d(A,\ B) = \frac{1}{2n}\sum_{j=1}^{n}\left(\left|\mu_A(x_j) - \mu_B(x_j)\right| + \left|\nu_A(x_j) - \nu_B(x_j)\right| + \left|\pi_A(x_j) - \pi_B(x_j)\right|\right)$$

Thus, for two IF numbers (IFNs) $\alpha = \langle \mu_\alpha,\ \nu_\alpha,\ \pi_\alpha\rangle$ and $\beta = \left\langle \mu_\beta,\ \nu_\beta,\ \pi_\beta\right\rangle$

$$d(\alpha,\ \beta) = \frac{1}{2}\left(\left|\mu_\alpha - \mu_\beta\right| + \left|\nu_\alpha - \nu_\beta\right| + \left|\pi_\alpha - \pi_\beta\right|\right) \tag{1}$$

Let m alternatives $A_i (i = 1,\ 2,\ \ldots,\ m)$ be evaluated relative to n criteria $H_j (j = 1,\ 2,\ \ldots,\ n)$ by an expert soliciting response as IVIFNs with TPCs producing decision matrix $D = \left(\alpha_{ij}\right)_{m\times n} = \left(\left\langle\left[\mu_{ij}^l,\ \mu_{ij}^u\right],\ \left[\nu_{ij}^l,\ \nu_{ij}^u\right],\ \left[\pi_{ij}^l,\ \pi_{ij}^u\right]\right\rangle\right)_{m\times n}$. Assume that the expert also evaluates criteria $H_j (j = 1,\ 2,\ \ldots,\ n)$ generating IVIF weight vector $\omega = \left(\omega_j\right)_{1\times n}$, where $\omega_j = \left\langle\left[\rho_j^l,\ \rho_j^u\right],\ \left[\sigma_j^l,\ \sigma_j^u\right],\ \left[\tau_j^l,\ \tau_j^u\right]\right\rangle$. Then, using Definition 2.3 and Eq. (12), let the crisp weights of criteria be $w = (w_1,\ w_2,\ \ldots,\ w_n)$.

Assume, without loss of generality, that $H_B = \{H_1,\ H_2,\ \ldots,\ H_s\}$ and $H_C = \{H_{s+1},\ H_{s+2},\ \ldots,\ H_n\}$, respectively, are the sets of benefit criteria and cost criteria. Then, the IVIF PIS (IVIFPIS), α^+, and IVIF NIS (IVIFNIS), α^-, are given by

$$\alpha^+ = \left(\alpha_1^+,\ \alpha_2^+,\ \ldots,\ \alpha_n^+\right), \text{where}\, \alpha_j^+ = \begin{cases} \langle [1,\ 1],\ [0,\ 0],\ [0,\ 0]\rangle, \text{for}\, j = 1,\ 2,\ \ldots,\ s \\ \langle [0,\ 0],\ [1,\ 1],\ [0,\ 0]\rangle, \text{for}\, j = s+1,\ s+2,\ \ldots,\ n \end{cases}$$

$$\alpha^- = \left(\alpha_1^-,\ \alpha_2^-,\ \ldots,\ \alpha_n^-\right), \text{where}\, \alpha_j^- = \begin{cases} \langle [0,\ 0],\ [1,\ 1],\ [0,\ 0]\rangle, \text{for}\, j = 1,\ 2,\ \ldots,\ s \\ \langle [1,\ 1],\ [0,\ 0],\ [0,\ 0]\rangle, \text{for}\, j = s+1,\ s+2,\ \ldots,\ n \end{cases}$$

Assume that $\mu_{ij} \in \left[\mu_{ij}^l,\ \mu_{ij}^u\right]$, $\nu_{ij} \in \left[\nu_{ij}^l,\ \nu_{ij}^u\right]$, $\pi_{ij} \in \left[\pi_{ij}^l,\ \pi_{ij}^u\right]$, then by Eq. (1), the weighted distance between an alternative $A_i (i = 1,\ 2,\ \ldots,\ m)$ and the IVIFPIS α^+ with respect to all the criteria $H_j (j = 1,\ 2,\ \ldots,\ n)$ is given by

$$d_i^+ = \sum_{j=1}^{n} w_j d\left(\alpha_{ij},\ \alpha_j^+\right) \tag{2}$$

$$\text{where}\quad d\left(\alpha_{ij},\ \alpha_j^+\right) = \begin{cases} \frac{1}{2}\left(\left(1 - \mu_{ij}\right) + \nu_{ij} + \pi_{ij}\right), \text{for}\, j = 1,\ 2,\ \ldots,\ s \\ \frac{1}{2}\left(\mu_{ij} + \left(1 - \nu_{ij}\right) + \pi_{ij}\right), \text{for}\ j = s+1,\ s+2,\ \ldots,\ n \end{cases}$$

$$\text{i.e.,} \quad d\left(\alpha_{ij},\ \alpha_j^+\right) = \begin{cases} 1-\mu_{ij},\ \text{for}\, j=1,\ 2,\ \ldots,\ s \\ 1-\nu_{ij},\ \text{for}\, j=s+1,\ s+2,\ \ldots,\ n \end{cases}$$

$$\text{where}\, \pi_{ij} = 1-\mu_{ij}-\nu_{ij}$$

By Eq. (1), the weighted distance between an alternative $A_i (i = 1,\ 2,\ \ldots,\ m)$ and the IVIFNIS α^- with respect to all the criteria $H_j (j = 1,\ 2,\ \ldots,\ n)$ is given by

$$d_i^- = \sum_{j=1}^{n} w_j d\left(\alpha_{ij},\ \alpha_j^-\right) \tag{3}$$

$$\text{where} \quad d\left(\alpha_{ij}, \alpha_j^-\right) = \begin{cases} \frac{1}{2}\left(\mu_{ij} + \left(1-\nu_{ij}\right) + \pi_{ij}\right),\ \text{for}\, j=1,\ 2,\ \ldots,\ s \\ \frac{1}{2}\left(\left(1-\mu_{ij}\right) + \nu_{ij} + \pi_{ij}\right),\ \text{for}\, j=s+1,\ s+2,\ \ldots,\ n \end{cases}$$

$$\text{i.e.,} \quad d\left(\alpha_{ij},\ \alpha_j^-\right) = \begin{cases} 1-\nu_{ij},\ \text{for}\, j=1,\ 2,\ \ldots,\ s \\ 1-\mu_{ij},\ \text{for}\, j=s+1,\ s+2,\ \ldots,\ n \end{cases}$$

By Eqs. (2) and (3), the relative closeness coefficients of alternative A_i are given by

$$\begin{aligned} C_i\left(\mu_{ij},\ \nu_{ij},\ \pi_{ij}\right) &= \frac{d_i^-}{d_i^- + d_i^+}, \quad \text{for}\, i=1,\ 2,\ \ldots,\ m \\ &= \frac{\sum_{j=1}^{s} w_j\left(1-\nu_{ij}\right) + \sum_{j=s+1}^{n} w_j(1-\mu_{ij})}{\left(\sum_{j=1}^{s} w_j\left(1-\nu_{ij}\right) + \sum_{j=s+1}^{n} w_j(1-\mu_{ij})\right) + \left(\sum_{j=1}^{s} w_j\left(1-\mu_{ij}\right) + \sum_{j=s+1}^{n} w_j(1-\nu_{ij})\right)} \\ &= \frac{\sum_{j=1}^{s} w_j\left(1-\nu_{ij}\right) + \sum_{j=s+1}^{n} w_j(1-\mu_{ij})}{\sum_{j=1}^{n} w_j\left(1+\pi_{ij}\right)} \\ &= \frac{P_i}{Q_i}\,(\text{say}), \quad \text{for}\, i=1,\ 2,\ \ldots,\ m \\ \text{Thus, } C_i &= \frac{P_i}{Q_i}, \quad \text{for}\, i=1,\ 2,\ \ldots,\ m \end{aligned} \tag{4}$$

Taking logarithm of both sides of (4) and on partial differentiations with respect to μ_{ij} $(j = 1,\ 2,\ \ldots,\ n)$, the following relations are obtained

$$\frac{\partial log C_i}{\partial \mu_{ij}} = \frac{w_j}{Q_i} > 0 \quad (j=1,\ 2,\ \ldots,\ s) \tag{5}$$

$$\frac{\partial log C_i}{\partial \mu_{ij}} = -\frac{w_j}{P_i} + \frac{w_j}{Q_i} \le -\frac{w_j}{P_i} + \frac{w_j}{P_i} = 0\left[\text{As}\, P_i \le Q_i\right]$$
$$(j = s+1,\ s+2,\ \ldots,\ n) \tag{6}$$

As taking logarithm does not change the monotonic behavior of a function, therefore, using Eqs. (5) and (6), $C_i\left(\mu_{ij},\ \nu_{ij},\ \pi_{ij}\right)$ is a monotonically increasing function

with respect to $\mu_{ij}(j = 1, 2, \ldots, s)$ and monotonically decreasing function with respect to $\mu_{ij}(j = s+1, s+2, \ldots, n)$.

Similarly, taking logarithm of both sides of Eq. (4) and partial differentiations with respect to $\nu_{ij}(j = 1, 2, \ldots, n)$, the following relations are formed

$$\frac{\partial logC_i}{\partial \nu_{ij}} = -\frac{w_j}{P_i} + \frac{w_j}{Q_i} \leq -\frac{w_j}{P_i} + \frac{w_j}{P_i} = 0\left[\text{As } P_i \leq Q_i\right] \ (j = 1, 2, \ldots, s) \tag{7}$$

$$\frac{\partial logC_i}{\partial \nu_{ij}} = -\frac{\left(-w_j\right)}{Q_i} = \frac{w_j}{Q_i} > 0 \quad (j = s+1, s+2, \ldots, n) \tag{8}$$

Thus, using Eqs. (7) and (8), $C_i\left(\mu_{ij}, \nu_{ij}, \pi_{ij}\right)$ is a monotonically decreasing function with respect to $\nu_{ij}(j = 1, 2, \ldots, s)$ and monotonically increasing function with respect to $\nu_{ij}(j = s+1, s+2, \ldots, n)$.

Since C_i depends on $\mu_{ij} \in \left[\mu_{ij}^l, \mu_{ij}^u\right]$, $\nu_{ij} \in \left[\nu_{ij}^l, \nu_{ij}^u\right]$, $\pi_{ij} \in \left[\pi_{ij}^l, \pi_{ij}^u\right]$, C_i is an interval $\left[C_i^l, C_i^u\right]$, where C_i^l and C_i^u, respectively, denote lower and upper bounds of C_i. The values of C_i^l and C_i^u are estimated by the following linear fractional programming models

$$c_i^u = max\left\{\frac{\sum_{j=1}^{s} w_j\left(1-\nu_{ij}\right) + \sum_{j=s+1}^{n} w_j(1-\mu_{ij})}{\sum_{j=1}^{n} w_j\left(1+\pi_{ij}\right)}\right\}$$

$$\text{subject to}\begin{cases} \mu_{ij}^l \leq \mu_{ij} \leq \mu_{ij}^u (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n) \\ \nu_{ij}^l \leq \nu_{ij} \leq \nu_{ij}^u (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n) \\ \pi_{ij}^l \leq \pi_{ij} \leq \pi_{ij}^u (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n) \end{cases} \tag{M-1}$$

and

$$C_i^l = min\left\{\frac{\sum_{j=1}^{s} w_j\left(1-\nu_{ij}\right) + \sum_{j=s+1}^{n} w_j(1-\mu_{ij})}{\sum_{j=1}^{n} w_j\left(1+\pi_{ij}\right)}\right\}$$

$$\text{subject to}\begin{cases} \mu_{ij}^l \leq \mu_{ij} \leq \mu_{ij}^u (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n) \\ \nu_{ij}^l \leq \nu_{ij} \leq \nu_{ij}^u (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n) \\ \pi_{ij}^l \leq \pi_{ij} \leq \pi_{ij}^u (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n) \end{cases} \tag{M-2}$$

The maximum value or upper bound C_i^u of C_i is found for

$$\mu_{ij} = \begin{cases} \mu_{ij}^{u}, \text{if} j = 1,\ 2,\ \ldots,\ s \\ \mu_{ij}^{l}, \text{if} j = s+1,\ s+2,\ \ldots,\ n \end{cases} \text{and } \nu_{ij} = \begin{cases} \nu_{ij}^{l}, \text{if } j = 1,\ 2,\ \ldots,\ s \\ \nu_{ij}^{u}, \text{if} j = s+1,\ s+2,\ \ldots,\ n \end{cases}$$

Again, the minimum value or lower bound C_i^l of C_i is attained for

$$\mu_{ij} = \begin{cases} \mu_{ij}^{l}, \text{ if} j = 1,\ 2,\ \ldots,\ s \\ \mu_{ij}^{u}, \text{ if} j = s+1,\ s+2,\ \ldots,\ n \end{cases} \text{and } \nu_{ij} = \begin{cases} \nu_{ij}^{u}, \text{if} j = 1,\ 2,\ \ldots,\ s \\ \nu_{ij}^{l}, \text{if} j = s+1,\ s+2,\ \ldots,\ n \end{cases}$$

Hence, the linear fractional programming models (M-1) and (M-2) yield the following simple results

$$C_i^u = \frac{\sum_{j=1}^{s} w_j\left(1-\nu_{ij}^{l}\right) + \sum_{j=s+1}^{n} w_j\left(1-\mu_{ij}^{l}\right)}{\sum_{j=1}^{s} w_j\left(2-\mu_{ij}^{u}-\nu_{ij}^{l}\right) + \sum_{j=s+1}^{n} w_j\left(2-\mu_{ij}^{l}-\nu_{ij}^{u}\right)} \tag{9}$$

$$C_i^l = \frac{\sum_{j=1}^{s} w_j\left(1-\nu_{ij}^{u}\right) + \sum_{j=s+1}^{n} w_j\left(1-\mu_{ij}^{u}\right)}{\sum_{j=1}^{s} w_j\left(2-\mu_{ij}^{l}-\nu_{ij}^{u}\right) + \sum_{j=s+1}^{n} w_j\left(2-\mu_{ij}^{u}-\nu_{ij}^{l}\right)} \tag{10}$$

Thus, the intervals of RCCs of m alternatives $A_i (i = 1,\ 2,\ \ldots,\ m)$ are given by $C_i = \left[C_i^l,\ C_i^u\right]$. "Alternative A_i is not dominated by A_k" is expressed by $A_i \succcurlyeq A_k$ whose possibility degree is measured by that of $C_i \geq C_k$. Thus, using Definition 2.2, the possibility degree of $A_i \succcurlyeq A_k$ is given by

$$p_{ik} = p(A_i \succcurlyeq A_k) = p(C_i \geq C_k) = min\left\{max\left\{\frac{C_i^u - C_k^l}{L(C_i) + L(C_k)},\ 0\right\},\ 1\right\} \tag{11}$$

where $C_i = \left[C_i^l,\ C_i^u\right]$, $C_k = \left[C_k^l,\ C_k^u\right]$, $L(C_i) = C_i^u - C_i^l$, and $L(C_k) = C_k^u - C_k^l$.

The additive reciprocal fuzzy preference relation is given by the possibility degree matrix $P = \left(p_{ij}\right)_{m\times m}$ which is obtained by pairwise comparison of intervals of RCCs $C_i (i = 1,\ 2,\ \ldots,\ m)$.

Xu and Da [22] advanced the optimal degree formula to compute crisp criteria weights or performance values of alternatives using possibility degree matrix $P = \left(p_{ij}\right)_{m\times m}$ which is given by

$$\xi_i = \frac{1}{m(m-1)}\left(\sum_{k=1}^{n} p_{ik} + \frac{m}{2} - 1\right), \text{ for } i = 1,\ 2,\ \ldots,\ m \tag{12}$$

3.1 Interval-Valued Intuitionistic Fuzzy TOPSIS Approach

On the basis of preceding deductions, the interval-valued intuitionistic fuzzy TOPSIS approach for dealing with MADM problems is proposed as follows:

Step 1. In a given MADM problem, n criteria $H_k(k = 1, 2, \ldots, n)$, divided into benefit criteria $H_B = \{H_1, H_2, \ldots, H_s\}$ and cost criteria $H_C = \{H_{s+1}, H_{s+2}, \ldots, H_n\}$ sets, and m alternatives $A_i(i = 1, 2, \ldots, m)$ are identified.
Step 2. IVIFN response α_{ik} with TPC by an expert for ith alternative relative to kth criterion gives the formation of decision matrix

$$D = (\alpha_{ik})_{m\times n} = \left(\left\langle\left[\mu_{ik}^l, \mu_{ik}^u\right], \left[\nu_{ik}^l, \nu_{ik}^u\right], \left[\pi_{ik}^l, \pi_{ik}^u\right]\right\rangle\right)_{m\times n}$$

Further, IVIFN response w_j with TPC by the expert for kth criterion gives weight vector $\omega = (\omega_k)_{1\times n} = \left(\left\langle\left[\rho_k^l, \rho_k^u\right], \left[\sigma_k^l, \sigma_k^u\right], \left[\tau_k^l, \tau_k^u\right]\right\rangle\right)_{1\times n}$.
Step 3. Using Definition 2.3, the additive reciprocal fuzzy preference relation $P = (p(\omega_i \geq \omega_k))_{n\times n} = (p_{ik})_{n\times n}$ from IVIF criteria weight vector $\omega = (\omega_k)_{1\times n}$ is obtained. By Eq. (12), the crisp criteria weight vector $w = (w_k)_{1\times n}$ with $w_k \in (0, 1]$ and $\sum_{k=1}^n w_k = 1$ is generated.
Step 4. The intervals of RCCs of m alternatives $A_i(i = 1, 2, \ldots, m)$ are given by $C_i = \left[C_i^l, C_i^u\right]$, where C_i^u and C_i^l are, respectively, given by Eqs. (9) and (10).
Step 5. Using Eq. (11), the possibility degree matrix $P = \left(p_{ij}\right)_{m\times m}$ for pairwise comparison of intervals of RCCs C_i of alternatives $A_i(i = 1, 2, \ldots, m)$ is obtained.
Step 6. The performance scores of alternatives $A_i(i = 1, 2, \ldots, m)$ are evaluated from $P = \left(p_{ij}\right)_{m\times m}$ using Eq. (12).
Step 7. The decreasing order of the optimal degrees ξ_i generates the ranking order of alternatives $A_i(i = 1, 2, \ldots, m)$.

4 An Illustrative Example

In this section, an example has been adapted from [17, 18, 24] for illustrating the application of the advanced TOPSIS approach in IVIF environment in MADM context. The problem is stated as follows.

An investment company plans to invest in four companies, viz. A_1(automobile), A_2(food), A_3(computer), and A_4(weapons). These alternatives are evaluated relative to three criteria, viz. H_1(risk assessment), H_2(growth assessment), and H_3 (environmental impact assessment), by an expert giving his/her opinion in IVIFNs yielding decision matrix as follows

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \end{array} \begin{array}{c} \begin{array}{ccc} H_1 & H_2 & H_3 \end{array} \\ \begin{pmatrix} \langle[0.4,\ 0.5],\ [0.3,\ 0.4],\ [0.1,\ 0.3]\rangle & \langle[0.4,\ 0.6],\ [0.2,\ 0.4],\ [0.0,\ 0.4]\rangle & \langle[0.1,\ 0.3],\ [0.5,\ 0.6],\ [0.1,\ 0.4]\rangle \\ \langle[0.6,\ 0.7],\ [0.2,\ 0.3],\ [0.0,\ 0.2]\rangle & \langle[0.6,\ 0.7],\ [0.2,\ 0.3],\ [0.0,\ 0.2]\rangle & \langle[0.4,\ 0.7],\ [0.1,\ 0.2],\ [0.1,\ 0.5]\rangle \\ \langle[0.3,\ 0.6],\ [0.3,\ 0.4],\ [0.0,\ 0.4]\rangle & \langle[0.5,\ 0.6],\ [0.3,\ 0.4],\ [0.0,\ 0.2]\rangle & \langle[0.5,\ 0.6],\ [0.1,\ 0.3],\ [0.1,\ 0.4]\rangle \\ \langle[0.7,\ 0.8],\ [0.1,\ 0.2],\ [0.0,\ 0.2]\rangle & \langle[0.6,\ 0.7],\ [0.1,\ 0.3],\ [0.0,\ 0.3]\rangle & \langle[0.3,\ 0.4],\ [0.1,\ 0.2],\ [0.4,\ 0.6]\rangle \end{pmatrix} \end{array}$$

The following two cases are considered depending on criteria weights, IVIFNs and crisp.

Case 1. *When criteria weights are IVIFNs with TPCs*
The TPC-based IVIFN criteria weight vector [17, 18] is $\omega = (\omega_k)_{1\times 3}$

$$= \begin{pmatrix} \langle[0.1,\ 0.4],\ [0.2,\ 0.55],\ [0.05,\ 0.7],\ [0.2,\ 0.5],\ [0.15,\ 0.45],\ [0.05,\ 0.65],\rangle \\ \langle[0.25,\ 0.6],\ [0.15,\ 0.38],\ [0.02,\ 0.6]\rangle \end{pmatrix}$$

Using Definition 2.3, the possibility degree matrix obtained by pairwise comparison of IVIFN criteria weights is

$$P = \left(p\left(\omega_i \geq \omega_j\right)\right)_{3\times 3} = \begin{pmatrix} 0.5 & 0.36 & 0.27 \\ 0.64 & 0.5 & 0.41 \\ 0.73 & 0.59 & 0.5 \end{pmatrix}$$

Using Eq. (12), the crisp criteria weight vector becomes $\mathrm{w} = (0.27,\ 0.34,\ 0.39)$.

Subcase 1. *Let* $H_B = \{H_1,\ H_2,\ H_3\}$, i.e., *all criteria are benefit criteria [17, 18]*
Using Eqs. (9) and (10), the intervals of RCCs of alternatives are obtained as

$$C_1 = [0.421,\ 0.547],\ C_2 = [0.607,\ 0.737],\ C_3 = [0.536,\ 0.660],\ C_4 = [0.610,\ 0.698]$$

Using Eq. (11), the possibility degree matrix for pairwise comparison of alternatives is given by

$$P = \left(p_{ij}\right)_{4\times 4} = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \end{array} \\ \begin{pmatrix} 0.5 & 0 & 0.044 & 0 \\ 1 & 0.5 & 0.791 & 0.583 \\ 0.956 & 0.209 & 0.5 & 0.236 \\ 1 & 0.417 & 0.764 & 0.5 \end{pmatrix} \end{array}$$

By Eq. (12), the performance scores of alternatives $A_i (i = 1,\ 2,\ 3,\ 4)$ as optimal degrees ξ_i are estimated as $\xi_1 = 0.129,\ \xi_2 = 0.323,\ \xi_3 = 0.242,\ \xi_4 = 0.307$.

Thus, the ordering of alternatives becomes $A_2 \succ A_4 \succ A_3 \succ A_1$ which is identical with that of Li [17, 18].

Subcase 2. *Let* $H_B = \{H_2\}$ *and* $H_C = \{H_1,\ H_3\}$, i.e., *growth assessment is benefit criterion while risk and environmental impact assessments are cost criteria.*

Using Eqs. (9) and (10), the intervals of RCCs of alternatives are obtained as

Table 1 Comparison of the proposed approach with other approaches

Approach	Criteria weights	Benefit criteria	Cost criteria	Ranking
Li [17]	IVIFN	H_1, H_2, H_3	–	$A_2 \succ A_4 \succ A_3 \succ A_1$
Li [18]	IVIFN	H_1, H_2, H_3	–	$A_2 \succ A_4 \succ A_3 \succ A_1$
Ye [24]	Crisp	H_1, H_2, H_3	–	$A_2 \succ A_4 \succ A_3 \succ A_1$
Proposed approach	IVIFN	H_1, H_2, H_3	–	$A_2 \succ A_4 \succ A_3 \succ A_1$
	Crisp	H_1, H_2, H_3	–	$A_2 \succ A_4 \succ A_3 \succ A_1$
	IVIFN	H_2	H_1, H_3	$A_1 \succ A_4 \succ A_3 \succ A_2$
	Crisp	H_2	H_1, H_3	$A_1 \succ A_3 \succ A_4 \succ A_2$

$$C_1 = [0.510,\ 0.634],\ C_2 = [0.383,\ 0.505],\ C_3 = [0.397,\ 0.521],\ C_4 = [0.419,\ 0.512]$$

Using Eqs. (11) and (12), the optimal degrees $\xi_i (i = 1,\ 2,\ 3,\ 4)$ are estimated as $\xi_1 = 0.371$, $\xi_2 = 0.195$, $\xi_3 = 0.215$, $\xi_4 = 0.220$ producing the ranking order $A_1 \succ A_4 \succ A_3 \succ A_2$.

Case 2. *When criteria weights are crisp*
Ye [24] assumed crisp criteria weight vector as w = (0.35, 0.25, 0.40).

Subcase 1. *Let* $H_B = \{H_1,\ H_2,\ H_3\}$, i.e., *all criteria are benefit criteria [24]*
Using Eqs. (9) and (10), the intervals of RCCs of alternatives are given by

$$C_1 = [0.419,\ 0.538],\ C_2 = [0.673,\ 0.737],\ C_3 = [0.529,\ 0.661],\ C_4 = [0.615,\ 0.700]$$

Using Eqs. (11) and (12), the performance scores of alternatives are estimated as $\xi_1 = 0.128$, $\xi_2 = 0.360$, $\xi_3 = 0.223$, $\xi_4 = 0.289$ generating the ranking order $A_2 \succ A_4 \succ A_3 \succ A_1$ which is identical with that of Ye [24].

Subcase 2. *Let* $H_B = \{H_2\}$ *and* $H_C = \{H_1,\ H_3\}$, i.e., *growth assessment is benefit criterion while risk and environmental impact assessments are cost criteria.*

Using Eqs. (9) and (10), the intervals of relative closeness coefficients of alternatives are given by

$$C_1 = [0.504,\ 0.621],\ C_2 = [0.351,\ 0.475],\ C_3 = [0.381,\ 0.512],\ C_4 = [0.385,\ 0.475]$$

By Eqs. (11) and (12), the optimal degrees are computed as $\xi_1 = 0.372$, $\xi_2 = 0.191$, $\xi_3 = 0.228$, $\xi_4 = 0.209$ giving the ranking $A_1 \succ A_3 \succ A_4 \succ A_2$.

The summary of the results appeared in the above example is given in Table 1.

Remark 5.1 From Table 1, following observations are made.

- If all criteria $H_1,\ H_2,\ H_3$ are benefit criteria with IVIFN weights [17, 18] and crisp weights [24], then all the rankings of alternatives based on the presented approach become $A_2 \succ A_4 \succ A_3 \succ A_1$ concurring with the rankings of both Li [17, 18] (for IVIFN criteria weights) and Ye [24] (for crisp criteria weights) validating the our method.
- If, further, the nature of criteria is considered as benefit criterion H_2 and cost criteria $H_1,\ H_3$, then the rankings $A_1 \succ A_4 \succ A_3 \succ A_2$ (for IVIFN criteria weights) and $A_1 \succ A_3 \succ A_4 \succ A_2$ (for crisp criteria weights) are a bit dissimilar and also differ from those of [17, 18, 24] which could be attributed to the complexity and nature of criteria weights.
- The proposed IVIF TOPSIS method is extensive in nature due to considering, both types of criteria, benefit and cost, with two types of data, IVIFN and crisp, but simple in formulation.
- The presented illustrations of the proposed approach validate its effectiveness due to easy computations involved and suitable for naive computer programming.

5 Concluding Remarks

As hesitancy degree has influence on distance measure in IF settings, the normalized hamming distance [21] containing three parameters is used in this study. The TOPSIS approach is extended in IVIF settings using mathematical programming techniques for dealing with MADM problems. The validity and effectiveness of the approach have been illustrated through the problems of Li [17, 18] and Ye [24] demonstrating the concurrence of all such rankings. The advanced method followed TPC of IFS in contrast to two parameter characterizations by Li [17, 18]. The computational complexity is minimized in the presented approach due to applications of its simple formulations. It is also found that the choices of criteria, benefit and cost, and types of criteria data, IVIFN and crisp, impact the ordering of alternatives based on the advanced TOPSIS method in IVIF settings. Hence, the presented approach has the power to model imprecision and uncertainty in more reliable and effective manner than the existing approaches. For future study, the TOPSIS can be extended in IVIF settings by using weighted absolute distance between IFSs in TPCs.

Acknowledgements The authors remain grateful to the reviewers for their valuable comments and suggestions to improve the quality of the manuscript.

References

1. Abo-Sinna, M.A., Amer, A.H.: Extensions of TOPSIS for multi-objective large-scale nonlinear programming problems. Appl. Math. Comput. **162**, 243–256 (2005)
2. Atanassov, K.T.: Intuitionistic fuzzy sets. Fuzzy Sets Syst. **20**(1), 87–96 (1986)
3. Atanassov, K.T., Gargov, G.: Interval-valued intuitionistic fuzzy sets. Fuzzy Sets Syst. **31**(3), 343–349 (1989)
4. Biswas, A., De, A.K.: An efficient ranking technique for intuitionistic fuzzy numbers with its application to chance constrained bilevel programming. Adv. Fuzzy Syst. **2016**, 1–12 (2016)
5. Biswas, A., Dewan, S.: Priority based fuzzy goal programming technique for solving fractional fuzzy goals by using dynamic programming. Fuzzy Inf. Eng. **4**, 165–180 (2012)
6. Biswas, A., Kumar, S.: An integrated TOPSIS approach to MADM with interval-valued intuitionistic fuzzy settings. Adv. Intell. Syst. Comput. **706**, 533–543 (2018)
7. Biswas, A., Adan, A., Halder, P., Majumdar, D., Natale, V., Randler, C., Tonetti, L., Sahu, S.: Exploration of transcultural properties of the reduced version of the morningness-eveningness questionnaire (rMEQ) using adaptive neuro fuzzy inference system. Biol. Rhythm Res. **45**(6), 955–968 (2014)
8. Biswas, A., Majumder, D., Sahu, S.: Assessing morningness of a group of people by using fuzzy expert system and adaptive neuro fuzzy inference model. Commun. Comput. Inf. Sci. **140**, 47–56 (2011)
9. Biswas, A., Modak, N.: A fuzzy goal programming technique for multiobjective chance constrained programming with normally distributed fuzzy random variables and fuzzy numbers. Int. J. Math. Oper. Res. **5**, 551–570 (2013)
10. Biswas, A., Modak, N.: Using fuzzy goal programming technique to solve multiobjective chance constrained programming problems in a fuzzy environment. Int. J. Fuzzy Syst. Appl. **2**, 71–80 (2012)
11. Chen, C.T.: Extension of the TOPSIS for group decision-making under fuzzy environment. Fuzzy Sets Syst. **114**(1), 1–9 (2000)
12. Chen, T.Y.: The inclusion-based TOPSIS method with interval-valued intuitionistic fuzzy sets for multiple criteria group decision making. Appl. Soft Comput. **26**, 57–73 (2015)
13. Debnath, J., Biswas, A., Presobh, S., Sen, K.N., Sahu, S.: Fuzzy inference model for assessing occupational risks in construction sites. Int. J. Ind. Ergon. **55**, 114–128 (2016)
14. Facchinetti, G., Ricci, R.G., Muzzioli, S.: Note on ranking fuzzy triangular numbers. Int. J. Intell. Syst. **13**, 613–622 (1998)
15. Hwang, C.L., Yoon, Y.: A state of the art survey. In: Multiple Attribute Decision Making: Methods and Applications. Springer, Berlin (1981)
16. Kumar, S., Biswas, A.: Use of possibility measures for ranking of interval-valued intuitionistic fuzzy numbers in solving multicriteria decision making problems. Commun. Comput. Inf. Sci. **776**, 153–167 (2017)
17. Li, D.F.: TOPSIS-based nonlinear-programming methodology for multi-attribute decision making with interval-valued intuitionistic fuzzy sets. IEEE Trans. Fuzzy Syst. **18**(2), 299–311 (2010)
18. Li, D.F.: Linear programming method for MADM with interval-valued intuitionistic fuzzy sets. Expert Syst. Appl. **37**, 5939–5945 (2010)
19. Mardani, A., Jusoh, A., Zavadskas, E.K.: Fuzzy multiple criteria decision-making techniques and applications—two decades review from 1994 to 2014. Expert Syst. Appl. **42**, 4126–4148 (2015)
20. Nadaban, S., Dzitac, S., Dzitac, I.: Fuzzy TOPSIS—a general view. Procedia Comput. Sci. **91**, 823–831 (2016)
21. Szmidt, E., Kacprzyk, J.: Distances between intuitionistic fuzzy sets. Fuzzy Sets Syst. **114**(3), 505–518 (2000)
22. Xu, Z.S., Da, Q.L.: A possibility based method for priorities of interval judgment matrices. Chin. J. Manag. Sci. **11**, 63–65 (2003)

23. Xu, Z., Yager, R.R.: Dynamic intuitionistic fuzzy multi-attribute decision making. Int. J. Approx. Reasoning **48**(1), 246–262 (2008)
24. Ye, J.: Multicriteria fuzzy decision-making method based on a novel accuracy function under interval-valued intuitionistic fuzzy environment. Expert Syst. Appl. **36**, 6899–6902 (2009)
25. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
26. Zeng, S., Xiao, Y.: TOPSIS method for intuitionistic fuzzy multiple-criteria decision making and its application to investment selection. Kybernetes **45**(2), 282–296 (2016)
27. Zhao, X.: TOPSIS method for interval-valued intuitionistic fuzzy multiple attributes decision making and its application to teaching quality evaluation. J. Intell. Fuzzy Syst. **26**(6), 3049–3055 (2014)

Computational Intelligence for Localization of Mobile Wireless Sensor Networks

Vaishali R. Kulkarni and Veena Desai

Abstract The localization of mobile nodes in wireless sensor networks has been formulated as a concave optimization problem. The same has been approached through biologically inspired firefly algorithm (FA) and the artificial bee colony (ABC) algorithm. In the proposed method, a mobile node approximates its distance from multiple anchor nodes. The distance and the coordinates of the anchors are the parameters used by FA and ABC algorithms for the accurate estimation of the location by minimizing the suitably defined localization error. The localization method used here is iterative, and it works in a distributed fashion. A comparison of the performances of FA and ABC algorithms in terms of localization accuracy and computation time has been presented. FA exhibits higher accuracy of localization, while ABC is quicker.

Keywords Wireless sensor networks · Localization · Swarm intelligence · Firefly algorithm · Artificial bee colony algorithm

1 Introduction

Wireless sensor networks (WSNs) consist of tiny, autonomous sensor nodes having limited memory and processing capacities. Sensors can sense, compute, share, and coordinate information from environment [1]. The sensed data can be atmospheric conditions, temperature, pressure, or the events such as the movement of animals, vehicles, etc. WSNs have been effectively used in applications including military surveillance, environmental and traffic monitoring, health care, pollution and crop monitoring. Sensors in WSNs may be deployed in a random manner using aircraft or

V. R. Kulkarni (✉)
Department of Computer Science and Engineering, M. S. Ramaiah University of Applied Sciences, Bengaluru, India
e-mail: vaishali@ieee.org

V. Desai
Department of Electronics and Communication Engineering,
KLS Gogte Institute of Technology, Belagavi, India
e-mail: veenadesai@git.edu

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_34

robots in unstructured environments, such as thick forest, underwater, buildings. The information collected by the nodes is meaningful only if their locations are known. Thus, the location of the sensor nodes is a critical issue in WSN applications.

Sensor nodes can get the knowledge about their location using localization process. In small and structured environment, localization can be performed manually using fixed devices at known locations. Manual method is infeasible in unstructured conditions and for large, scalable WSNs. Installation of global positioning system (GPS) on sensor nodes is another commonly used localization technique. However, this method suffers from the requirement of additional hardware and increased cost. Also, GPS fails to reach the satellite in harsh environments as tall heritage buildings, thick canopy, underground areas, or moving objects. With these constraints, GPS can be installed on limited sensor nodes. These nodes are referred as landmarks, beacons, or anchors. Nodes which are not aware of their location are referred as unknown nodes or dumb nodes. Anchors broadcast their location to localize unknown nodes. Unknown nodes measure the distance between themselves and beacons using communication signals. The signals may be measured in the form of time difference of arrival (TDoA), received signal strength indication (RSSI), angle of arrival (AoA), time of arrival (ToA) etc. [2]. Measurement-based localization techniques are used in range-based localization. The measured values are then manipulated using localization techniques to estimate the location. Conventional localization techniques are trilateration, multilateration triangulation, bounding box etc. [3]. There exist range-free localization techniques which depend on the relative connectivity of the nodes. Distance vector (DV)-based localization, approximation algorithms, centroid algorithms are some of the examples of range-free localization techniques. Aforementioned localization techniques can be further classified as centralized or distributed, coarse-grained or fine-grained, anchor-free or anchor-based, deterministic or heuristic [4].

Mobility in WSNs has attracted the attention of researchers in recent years. Applications such as vehicle and packages monitoring, animal movement tracking, health monitoring are running over a period of time, and results are obtained using the current state of a phenomenon. The scenarios in mobile WSNs can be classified as mobile beacons and stationary unknown node; mobile unknown nodes and static beacons; and mobile beacons and mobile unknown nodes [5]. There are many advantages of mobile sensor nodes such as use of minimum beacons, reliable connection, optimal path plan by mobile anchors, improvement in the speed, and accuracy of localization [6]. Mobility is advantageous but also brings several challenges. Mobile sensor nodes change their position dynamically, and it is necessary to estimate the sensor positions more frequently. Localizing sensor in movement brings the challenge of connectivity, security, and energy consumption [7]. This demands for a simple, robust, and flexible localization methods. The mobile sensor localization algorithms must be autonomous and capable of handling the uncertainties, link failures, node mobility in network. Computational intelligence (CI) has algorithms that have such qualities. CI techniques can work in the environment of imprecision and uncertainty. CI includes techniques that are adaptive. These techniques can behave intelligently in complex environments. Some of the important paradigms of CI are metaheuristic

and swarm intelligence (SI), neural networks, fuzzy logic (FL), and evolutionary algorithms [8].

A distributed localization of mobile nodes using biologically inspired heuristic algorithms namely firefly algorithm (FA) and the artificial bee colony (ABC) algorithm is presented in this paper. Both the algorithms are based on SI paradigm that mimics the social behavior of biological species. FA is based on the bioluminescence signaling behavior of fireflies, and it provides a good balance of exploration versus exploitation [9]. ABC is a metaheuristic algorithm based on intelligent foraging behavior of honeybees [10]. The proposed work presents the comparison of the algorithms for accuracy in location estimation and computing time for mobile nodes in WSN. The primary contributions of this paper are:

1. The mobile node's current location estimation has been performed using an optimized localization algorithm.
2. The FA has been investigated for localization error minimization.
3. For each movement of the sensor, the details of the numeric results have been presented in terms of accuracy and computing time.
4. The simulation results have been presented.
5. A comparison of the results of FA-based localization and the localization using ABC algorithm has been presented.

The rest of this paper has been organized as follows: In Sect. 2, previous research in mobile sensor localization and the applications of stochastic algorithms has been discussed. Section 3 presents a model of localization of a sensor node in a WSN mission field. The FA algorithm has been outlined in Sect. 4. Section 5 presents the details of numerical simulation and the results. Section 6 presents the comparison of FA-based and ABC-based localization. Finally, Sect. 7 presents the conclusion and scope for future research.

2 Related Work

The architecture of mobile sensor networks can be classified as flat, 2−tier, or 3−tier hierarchical. These architectures vary in terms of running multiple heterogeneous networks. A survey of mobile sensor localization conducted with measurement-based techniques and assistance of anchors has been presented in article [6].

Unlike static WSNs, the mobile sensor nodes need to estimate their location in a periodic manner. As the localization is more frequent in comparison with static nodes, the centralized localization is not suitable for mobile WSNs because of delay and additional cost involved in transferring the data between nodes and central processing station. Therefore, distributed localization algorithms have been popular in mobile WSNs in which independent nodes estimate their location with accuracy in a reasonable time. Mobile anchors help in reducing cost design of WSN as minimum number of beacons follows a critical path to localize unknown nodes. Use of mobile anchors makes the network scalable, and there have been several distance

vector (DV) hop algorithms for finding the critical path for a mobile anchor [11]. Localization with optimization of anchor positions has been introduced in article [12].

When the anchors are mobile, the distance measurement from unknown nodes may not be precise. Mobile node localization with a focus on reliable distance measurement between beacon and a sensor node has been presented in article [13]. Conventional localization techniques estimate the location of unknown nodes using anchor location and distances from them as input to mathematical techniques. When the distance measurement is erroneous due to environmental noise or defects in the hardware used, the location estimation is inaccurate. There is a need of optimization of the localization error. With this, the localization problem has been formulated as an error minimization problem and many CI-based techniques have been successfully applied in localization. Fuzzy logic rules have been used for heuristic localization of a mobile sensor node [14]. The performance of the mobile WSN can be improved with optimal mobility of the nodes. In the some of the target tracking applications of WSNs, if the mobility is controlled, it is desirable to predict the movement of the nodes. A novel approach that performs location estimation with future path prediction and optimization of beacon movement has been presented in article [15]. The optimal sensor deployment and scheduling of sensors ensure efficient coverage and working of WSNs. The ABC algorithm and particle swarm optimization (PSO) for sensor deployment and scheduling have been proposed in article [16].

Range-free localization is coarse-grained and estimates the locations approximately by using the information from neighboring nodes. A range-free localization method that uses mobile anchor and neighbors has been proposed in article [17]. The estimated results are then optimized to reduce the localization error using fish swarm optimization algorithm (FSO). The localization of the sink node influences the energy consumption and lifetime of WSNs. A sink node localization approach based on grey wolves optimization algorithm has been proposed in article [18].

There have been several localization techniques for path planning and optimization of the path for mobile anchor. These algorithms mainly concentrate on localizing as many nodes as possible with least error. There are numerous localization algorithms that are proposed to handle the challenges as obstacles, scalability, security, non-line of sight, environmental noise, movement of sensors in WSNs, etc. The key metrics used in evaluating localization techniques include number of localized nodes, accuracy in location estimation and computing time, energy consumption, and scalability.

3 Proposed Work: Mobile Node Localization

The sensor field used in this work is two-dimensional and in the area of 100×100 square units. There are A anchor nodes with a communication range of R units. Anchor nodes are termed as $a_1, a_2, \ldots, a_A$. The target nodes are the mobile dumb nodes denoted by $m_1, m_2, \ldots, m_M$. This is shown in Fig. 1. Each

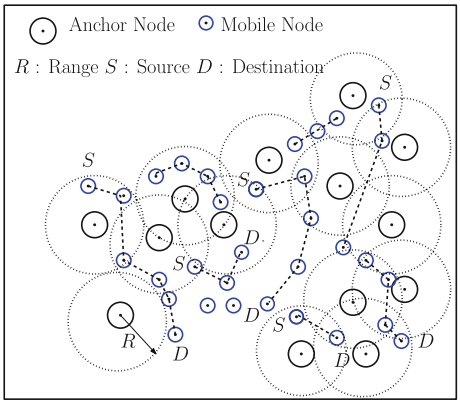


Fig. 1 WSN with mobile nodes and anchors

m changes its position, and the changing locations of the M nodes are represented as (m_{1x}, m_{1y}), (m_{2x}, m_{2y}), …, (m_{sx}, m_{sy}). Anchors have prior information of their location either using manual method or with the installation of GPS. A mobile dumb node is unaware of its location. Anchor node locations are represented as (a_{1x}, a_{1y}), (a_{2x}, a_{2y}), …, (a_{Ax}, a_{Ay}). The aim of the localization is to determine the multiple locations of M target sensor nodes that change their position s times. The location of each mobile dumb node position is expressed as m_{jx}, m_{jy}, and anchor node position is represented as a_{ix}, a_{ix}.

An unknown mobile node can be localized if it is in the vicinity of three non-collinear anchor nodes. The trilateration method that uses three non-collinear anchors has been used here [19]. The mobile node measures the parameters such as RSSI or TDoA or ToA and estimates its distances from anchors using (1). The distance measured is erroneous due to the environmental noise. This error is added using additive noise as given in (4). Therefore, the distance is estimated as $\hat{d}_i = d_i + f$. The term d_i is computed using (1), and f is the additive noise.

$$d_i = \sqrt{(a_{ix} - m_{jx})^2 + (a_{iy} - m_{jy})^2} \tag{1}$$

Equation (2) has been used to produce the additive noise f.

$$f = r_1 \left(\frac{e}{100}\right) (-1)^{r_2} \tag{2}$$

where r_1 parameter is a random number of uniform distribution in the range [0, 1] and r_2 is selected as 0 or 1 in a random manner.

The three non-collinear anchor locations and the measured distance are given as input to the FA optimization algorithm. The algorithm substitutes random locations (m_x, m_y) in the fitness function expression (3) so that the mean localization error e is minimum.

$$e_{jl} = \frac{1}{3}\sum_{i=1}^{3}\left(\sqrt{(m_{jx} - a_{ix})^2 + (m_{jy} - a_{iy})^2} - \hat{d}_i^{\,2}\right) \tag{3}$$

The algorithm returns the pair (m_x, m_y) which has resulted in lowest e in $k_{\max}$ cycles. The FA algorithm uses population of fireflies denoted as S for substitution of random locations in $k_{\max}$ iterations. The algorithm minimizes e and returns the position $(\hat{m}_{ix}, \hat{m}_{iy})$. In this way, the location of a dumb node for its every movement is estimated. The localization error for the localized node is given in (4).

$$E = \sqrt{(m_{ix} - \hat{m}_{ix})^2 + (m_{iy} - \hat{m}_{iy})^2} \tag{4}$$

The nodes that are not in the range of the three non-collinear anchors remain unlocalized. The pseudocode of the mobile node localization algorithm has been given in Algorithm 1.

Algorithm 1 Localization algorithm

```
Initialize anchor positions, communication range and percentage of noise
M = Number of dumb nodes and s = count for mobile positions
while i ≠ M do
  Generate s randomly
  for j ≤ S do
    Calculate d using (1) for the first location of m_i
    if m_i is within the range of 3 anchors then
      Pass the anchor positions and d to the FA algorithm
      The algorithm uses (3) and returns (m̂_ix, m̂_iy)
    end if
    j = j + 1
  end for
  i = i + 1
end while
Find the localization error using (4)
```

4 The Firefly Algorithm

The FA is based on swarm of fireflies that use short and rhythmic flash for attracting each other. The algorithm assumes all the fireflies as unisex, and they are attracted to each other based on their brightness. Therefore, the flies with better brightness attract the one with less brightness intensity. The brightness of a firefly decreases as the distance increases. This in turn reduces the attraction among them. FA has been used in optimization problem where the brightness of the fly determines the fitness. The intensity of brightness is varied, and new flies are attracted to search new locations.

The fireflies in FA are mobile and have adjustable visibility. With these features, there is more versatility in attractiveness variations that helps in exploring the search space in an effective manner. The variation of light intensity and formulation of the

attractiveness are the two important factors in FA. The stepwise details of firefly algorithm are given below:

Calculation of attraction: The intensity of the brightness is expressed as I, and the locations are denoted as $X_i = X_1, X_2, \ldots, X_{nd}$, where n is the number of flies and d is the dimension. The attractiveness is termed as β. The value of β varies with the variation in the distance r_{ij} between the firefly i and j and the absorption of light in the media. The absorption coefficient is termed as γ. The light intensity varying at source I_S as per the inverse law can be expressed using (5).

$$I(r) = \frac{I_s}{r^2} \tag{5}$$

With a fixed γ, I can be expressed using (6).

$$I = I_0 \mathrm{e}{-}\gamma r \tag{6}$$

where I_0 is the light intensity at origin. Gaussian form can be used to achieve the combined effect of both the inverse square law and absorption as expressed in (7).

$$I(r) = I_0 \mathrm{e}^{-\gamma r^2} \tag{7}$$

The fireflies' attractiveness β is proportional to the light intensity seen by adjacent fireflies. Thus, β can be expressed using (8).

$$\beta(r) = \beta_0 \mathrm{e}^{-\gamma r^2} \tag{8}$$

The parameter γ is important as it determines the speed of convergence of the FA.

Distance Measurement: The distance between two fireflies is measured as the cartesian distance as expressed in (9).

$$r_{ij} = ||X_i - X_j|| = \sum_{k=1}^{d} (X_{i,k} - X_{j,k})^2 \tag{9}$$

where x_i and x_j are any two fireflies. $X_{i,k}$ is the kth component of the spatial coordinate X_i of the ith fly.

Movement toward brighter firefly: The firefly movement toward better position is computed using initial position, attraction component, and the randomization parameter as expressed in (10).

$$X_i = X_i + \beta_0 \mathrm{e}^{-\gamma r_{i,j}^2} (X_j - X_i) + \alpha(r_1 - 0.5) \tag{10}$$

where X_i is the original position, the attraction is denoted by the second term, and the third term uses α as a randomization parameter. r_0 is a random number uniformly distributed in the range [0, 1].

Pseudocode for FA: α, β, and γ represent the radius of random solutions, step size toward better solution, and absorption coefficient, respectively. There can be two special cases for FA when there cannot be exploitation and exploration of positions. These cases are as given below:

1. When the weather is completely clear and no light dispersion, then each firefly can see each other. In this case, $\gamma = 0$, then $\beta(r) = \beta_0$
2. When there is a foggy weather with highest light dispersion, then fireflies cannot see each other. In this case, $\gamma = \infty$ and $\beta = 0$
   The firefly uses random positions during the above-mentioned cases. The values of β and α can be set to random positions in order to avoid extreme cases. The pseudocode for FA has been presented in Algorithm 2.

Algorithm 2 The pseudocode of the FA

```
Initialize X_i randomly such that x_min ≤ x_i ≤ x_max and i : 1, 2, 3, ..., n;
Evaluate intensity I_ij using the objective function F_x
Define γ
while (k ≤ k_max) do
  for i = 1 to n do
    for j = 1 to i do
      Evaluate I_i and I_j using F_x
      Compare fitness and exchange for better I
      Update I using (7)
       Compute β using (8)
       Generate new X_i using (10)
       j = j + 1
    end for
    i = i + 1
  end for
  Find the best solution g with highest I
  k = k + 1;
end while
```

The FA has been used in various optimization techniques in science and engineering. Some of the areas of FA applications include image processing, scheduling, clustering, fault detection, design and dynamic problems. A DV-Hop localization algorithm using firefly optimization has been presented in [20]. The use of FA has shown remarkable reduction in average localization error than the original algorithm. In an article [21], a node localization algorithm called parallel firefly algorithm (PFA) has been presented with improvement in objective function. PFA has been compared with PSO and GA, and it is proved that the performance of PFA is better. Article [22] presents the use of FA for routing in WSNs. The research uses FA to optimize the routing algorithm based on node degree, residual energy, and the distance.

The modified FA-based routing has been compared with energy-aware distributed clustering algorithm and distributed heterogeneous cluster-based routing. The FA-based algorithms perform better than the conventional routing algorithm.

5 Numerical Simulations and Results

FA-based mobile sensor node localization proposed here has been validated through MATLAB simulations. The parameters used in sensor field are:

1. Communication range $R = 40$ units;
2. Deployment field area = 100×100 square units;
3. Percentage of error $P = 1$ units;
4. Number of dumb nodes $M = 10$;
5. Number of beacons $A = 4$;
6. Mobility count of each dumb node = C where C is a random integer in the range [1, 10].

The parameter used in FA are:

1. Population of fireflies $S = 100$;
2. Dimension $D = 2$;
3. Maximum iterations $(k_{\max}) = 10$;
4. Attraction coefficient base value $B = 1$;
5. Light absorption coefficient $G = 1$;
6. Random coefficient $A_l = 0.3$.

Table 1 Localization of 3 mobile node using FA with $C = 6$, $R = 40$, and $P = 1$

Node 1			Node 2			Node 3		
Q_j	Actual location	Estimated location	Q_j	Actual location	Estimated location	Q_j	Actual location	Estimated location
	(x, y)	$(\hat{x}, \hat{y})$		(x, y)	$(\hat{x}, \hat{y})$		(x, y)	$(\hat{x}, \hat{y})$
1	(81.92, 7.84)	(81.92, 7.79)	1	(34.26, 24.96)	(34.13, 24.98)	1	(37.811, 54.90)	(37.54, 54.93)
2	(18.46, 46.95)	(18.98, 46.27)	2	(95.66, 28.03)	(95.544, 27.88)	2	(61.12, 61.34)	(61.17, 61.40)
3	(70.93, 45.46)	(70.842, 45.33)	3	(59.66, 17.98)	(59.42, 18.78)	3	(99.13, 71.18)	(99.36, 61.40)
4	(32.93, 7.11)	(32.33, 7.47)	4	(23.90, 19.10)	(23.58, 18.6)	4	(73.70, 89.20)	(73.75, 89.03)
5	(66.92, 1.25)	(66.46, 1.48)	5	(65.4, 73.02)	(65.40, 72.48)	5	(71.13, 73.79)	(71.18, 73.67)
6	(49.39, 37.37)	(49.20, 37.05)	6	(29.09, 56.49)	(28.87, 56.30)	6	(88.28, 10.28)	(88.92, 10.92)

Q_j denotes the jth movement. $j = 1, 2, \ldots, C$

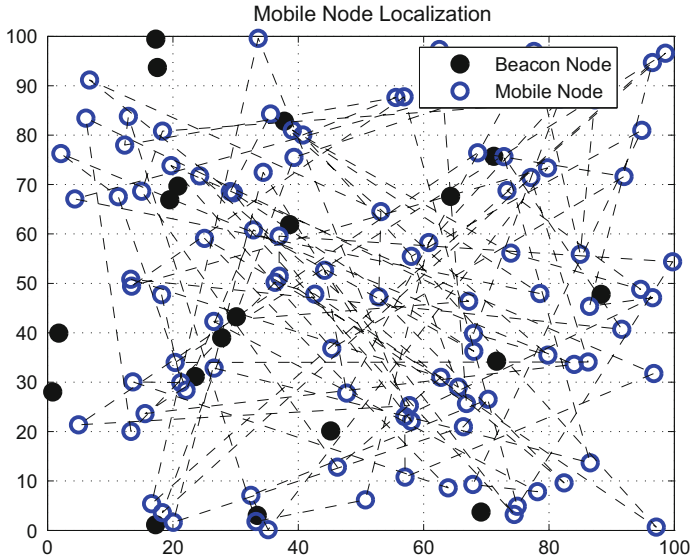


Fig. 2 WSN with mobile nodes

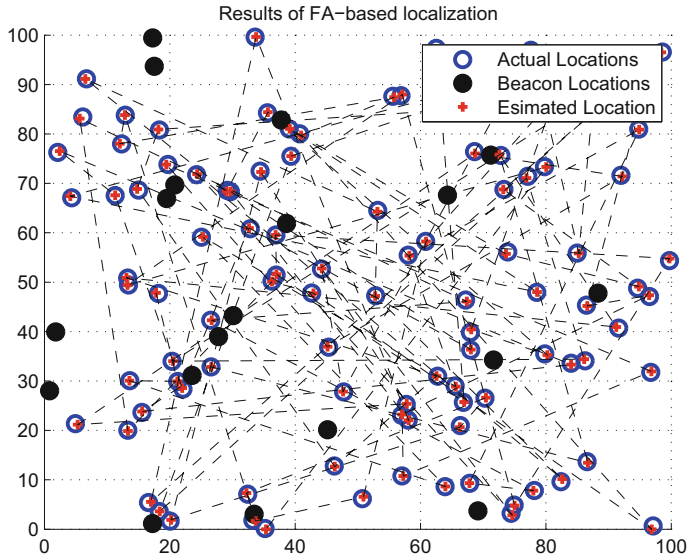


Fig. 3 FA-based localized WSN

The algorithm is tested on 10 mobile nodes with 4 anchor nodes. The sample results of 3 mobile nodes moving at 6 random positions have been presented in Table 1. Parameter Q indicates the index for mobile position of M_i node, where $i = 1, 2, \ldots, 10$.

Table 2 Statical summary of 10 mobile node localization using FA

Mobile node (M_i)	Mobility count (C)	Error (E)	Time (T)
1	4	0.0033	8.11
2	8	0.7270	28.26
3	10	0.0246	45.23
4	7	0.4938	20.35
5	6	0.2601	23.13
6	5	0.1369	13.32
7	9	0.0184	41.14
8	4	0.0351	8.46
9	3	0.7001	7.05
10	6	0.3636	22.80

The FA-based localization is further experimented on 100 moving sensor nodes with C times movement. With this, the locations to be determined are: $(\hat{x}, \hat{y}) = C \times 100$. Figure 2 depicts the movement of 100 mobile sensor nodes in a 100×100 square units mission space. Thirty beacons are deployed at known position. FA optimizes the E and estimates the location accurately. FA has been used for $k_{\max}$ iterations with N population of random locations to minimize the localization error. Figure 3 depicts the final stage of the localization with actual locations and estimated locations. Localization error has been expressed as E, and computing time has been denoted by T. The FA algorithm has been tested for 10 iterations. The mean of 10 trial runs for E and T for this scenario is represented in Table 2.

6 Comparison with the ABC-Based Localization

The results of the FA-based localization have been compared with another nature-inspired metaheuristic ABC algorithm. The ABC algorithm is based on swarm of bees searching for food sources. Bees are classified as employed bees, onlookers, and scouts. The food sources represent the possible solutions. Employed bees search for food sources and share the details with the other groups of bees. Based on the quality of the solutions, the onlookers change their locations using a greedy approach. If there is no improvement in the fitness in a given number of iterations, then the scout bees are generated in a random manner. These steps are repeated until the optimum solution is reached or the maximum iterations are completed. The critical steps in ABC are:

1. Initialization of employed bees: Bees are initialized as $x_{1D}, x_{2D} \ldots x_{sD}$ with lower limit and upper limit as $x_{\min}, x_{\max}$.
2. Onlooker bees generation : Onlooker bees are assigned to food locations using the probability of better fitness. The onlooker locations are generated with greedy

Table 3 Mean of E and T for mobile nodes with $S = 10$

Trial	Localization error E		Computing time T	
#	ABC	FA	ABC	FA
1	0.0567	0.0526	1.2315	13.9564
2	0.0276	0.0133	3.2432	65.2511
3	0.1322	0.0611	2.2107	14.7358
4	0.0266	0.0272	1.3254	52.9004
5	0.0732	0.0554	1.2876	34.0468
6	0.0669	0.0128	4.3214	60.2074
7	0.072	0.0476	2.1245	49.0849
8	0.096	0.0129	3.4216	42.7529
9	0.3382	0.1338	1.8923	22.0933
10	0.233	0.0421	2.0237	75.5515

approach from the existing memory using (11).

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{wj}) \tag{11}$$

Here $i, w \in 1, 2, 3, \ldots, s$ and $j = 1, 2, \ldots, D$. Index w is randomly chosen such that $w \neq i$ and $-1 \leq \phi_i \leq 1$.

3. Scout bees development: When there is no improvement in the solutions with employed and onlookers, a limit parameter L is used to abandon the food source. This is done when the algorithm completes $k_{\max}$ iterations. Scout bees are generated randomly to replace existing food positions as given in (12), where z is a random number, in the range [0, 1].

$$x_{ij} = x^j_{\min} + z(x^j_{\max} - x^j_{\min}) \tag{12}$$

The parameters used in ABC-based mobile node localization are: Kulkarni et al. [23].

1. Bees population $s = 100$;
2. Limit for elimination of bees $L = s \times D$;
3. Maximum iterations $(k_{\max}) = 100$.

ABC- and FA-based mobile node localization is performed for 100 dumb nodes and 20 beacons. For the sake of statistics, the dumb nodes are moved for $S = 10$ and the mean of E and T is recorded as presented in Table 3. To determine average and standard deviation in localization error and computing time of FA- and ABC-based localization, the mobility parameter C is restricted to smaller number of 3 count. For the faster communication, the communication radius is also enhanced to 50 units. The summary of the results of 10 trials is given in Table 4. Based on results, it is observed that the FA shows superiority in accuracy of locations than the ABC algorithm. But the ABC algorithm outperforms the FA in terms of the computing time. The ABC algorithm includes a control parameter that generates scout bees for

Table 4 Statistical summary of 50 trial runs of FA- and ABC-based WSN node localization

	Localization error		Computing time	
	Mean	Standard deviation	Mean	Standard deviation
FA	**0.0231**	**0.1531**	45.1417	0.3163
ABC	0.0102	0.1528	**10.1324**	**0.1121**

For all trials, $P = 1$, $A = 20$ and $M = 100$

generating random location when there is no improvement in the fitness. This feature of ABC improves the speed of convergence.

There are fixed parameters in FA. FA finds the optimum solution in each iteration and does not compare it with the previously obtained values. This results in inferior solution, and the fitness improvement does not take place. With this, FA generates random solutions. FA local search gets initialized during every iteration, and this repetitive procedure causes the delay. A modification of remembering the past better solutions can improve the speed of convergence with quality. The ABC algorithm is more flexible and uses few control parameters, with which it leads to faster convergence. The results obtained using the ABC algorithm are inferior than the FA-based localization. The choice between these two algorithms depends on the type of WSN application where the attention is on the quality or the speed of localization.

The accuracy of the estimated location depends on the percentage of noise associated with measured distance between anchors and dumb node. Figure 4 depicts the change in localization error with the increase in percentage of noise.

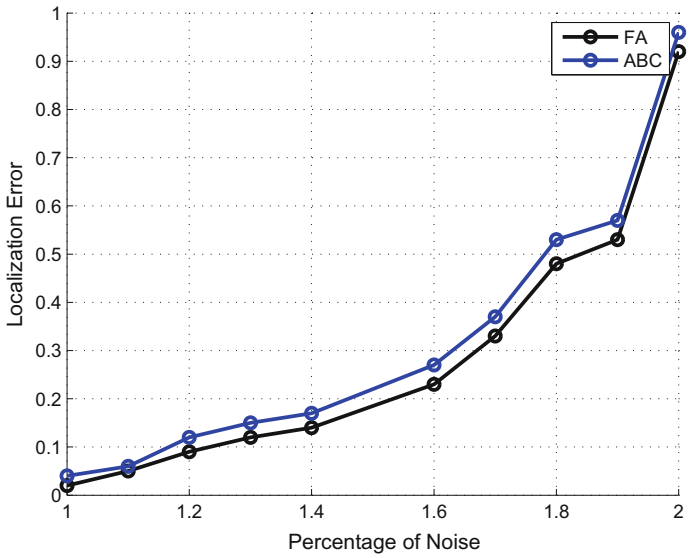


Fig. 4 Variation of the localization error based on the percentage of noise

7 Conclusion

In this research, SI algorithms have been investigated for the localization of mobile sensor nodes that are deployed in noisy environment. The localization is performed in a distributed manner by each moving dumb node at each of its movement point of journey. FA and ABC algorithms have been applied for minimization of localization error. A localization algorithm has been presented for estimating the locations of the mobile sensor node. The details of the numeric simulations and the results have been presented. The results of FA-based localization have been compared with those of the ABC algorithm. Simulation results show that the ABC algorithm requires less time for localization, whereas the FA performs better in terms of accuracy. The optimized mobile WSNs used in this work can enhance the efficiency of WSN. When sensors are dropped randomly, they may not guarantee connected networks, but mobile sensors may connect separated component and provide better service. With optimized localization of mobile sensors, more sensors can be moved closer to the event locations with complete coverage of the field. In case there is bottleneck of sensor nodes, it can be resolved using mobility.

This research can be further extended to optimize the energy during localization of mobile WSNs. Further, this study aims at localization using mobile beacons which can minimize the density of beacons, thereby reducing the costs of WSNs by maintaining the accuracy. Use of heuristic algorithms for optimal path planning of mobile beacon may be another direction of the future research. Also, the scope of this research is to develop variant of the FA and the ABC algorithms for the enhancement of speed and quality of solution.

References

1. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2008)
2. Boukerche, A., Oliveira, H.A.B., Nakamura, E.F., Loureiro, A.A.F.: Localization systems for wireless sensor networks. IEEE Wirel. Commun. Mag. **14**(6), 6–12 (2007)
3. Mao, G., Fidan, B., Anderson, B.D.O.: Wireless sensor network localization techniques. Comput. Netw. **51**(10), 2529–2553 (2007)
4. Cheng, L., Wu, C., Zhang, Y., Wu, H., Li, M., Maple, C.: A survey of localization in wireless sensor network. IJDSN **8** (2012)
5. Hu, L., Evans, D.: Localization for mobile sensor networks. In: 10th Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 45–57. ACM (2004)
6. Amundson, I., Koutsoukos, X.D.: A Survey on Localization for Mobile Wireless Sensor Networks, pp. 235–254. Springer, Berlin, Heidelberg (2009)
7. Halder, S., Ghosal, A.: A survey on mobility-assisted localization techniques in wireless sensor networks. J. Netw. Comput. Appl. **60**, 82–94 (2016)
8. Engelbrecht, A.P.: Computational Intelligence: an introduction, 2nd edn. Wiley, New York, USA (2007)
9. Yang, X.S.: Firefly Algorithm, levy flights and global optimization. ArXiv e-prints (2010)
10. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)

11. KUANG, X.H., Shao, H.: Distributed localization using mobile beacons in wireless sensor networks. J. China Univ. Posts Telecommun. **14**(4), 7–12 (2007)
12. Tuba, E., Tuba, M., Simian, D.: Range based wireless sensor node localization using bat algorithm. In: Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN), pp. 41–44. Malta (2016)
13. Kim, E., Kim, K.: Distance estimation with weighted least squares for mobile beacon-based localization in wireless sensor networks. IEEE Signal Process. Lett. **17**(6), 559–562 (2010)
14. Chiang, S.Y., Wang, J.L.: Localization in Wireless Sensor Networks by Fuzzy Logic System, pp. 721–728. Springer, Berlin Heidelberg (2009)
15. Mourad, F., Chehade, H., Snoussi, H., Yalaoui, F., Amodeo, L., Richard, C.: Controlled mobility sensor networks for target tracking using ant colony optimization. IEEE Trans. Mobile Comput. **11**(8), 1261–1273 (2012)
16. Mini, S., Udgata, S.K., Sabat, S.L.: Sensor deployment and scheduling for target coverage problem in wireless sensor networks. IEEE Sens. J. **14**(3), 636–644 (2014)
17. Sivakumar, S., Venkatesan.: Error minimization in localization of wireless sensor networks using fish swarm optimization algorithm. Int. J. Comput. Appl. **159**(7), 39–45 (2017)
18. Fouad, M.M., Hafez, A.I., Hassanien, A.E., Snasel, V.: Grey wolves optimizer-based localization approach in WSNS. In: 11th International Computer Engineering Conference (ICENCO), pp. 256–260 (2015)
19. Patwari, N., Ash, J.N., Kyperountas, S., Hero, A.O., Moses, R.L., Correal, N.S.: Locating the nodes: cooperative localization in wireless sensor networks. IEEE Signal Process. Mag. **22**(4), 54–69 (2005)
20. Pei, B., Zhang, H., Pei, T., Wang, H.: Firefly algorithm optimization based WSN localization algorithm. In: International Conference on Information and Communications Technologies (ICT 2015), pp. 1–5 (2015)
21. Sai, V.O., Shieh, C.S., Nguyen, T.T., Lin, Y.C., Horng, M.F., Le, Q.D.: Parallel firefly algorithm for localization algorithm in wireless sensor network. In: 3rd International Conference on Robot, Vision and Signal Processing (RVSP), pp. 300–305 (2015)
22. Lalwani, P., Ganguli, I., Banka, H.: FARW: firefly algorithm for routing in wireless sensor networks. In: 3rd International Conference on Recent Advances in Information Technology (RAIT), pp. 248–252 (2016)
23. Kulkarni, V.R., Desai, V., Kulkarni, R.V.: Multistage localization in wireless sensor networks using artificial bee colony algorithm. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8 (2016)

Reasoning for Uncertainty and Rough Set-Based Approach for an Efficient Biometric Identification: An Application Scenario

Ajeet Singh, Vikas Tiwari, Priyanka Garg and Appala Naidu Tentu

Abstract In the theory of knowledge discovery, two fundamental concepts are classification and categories. Some of the specific categories may be definable inside one set of knowledge, but they may be undefinable into another knowledge base. This paper contains significant rough membership functional properties, which are utilized in approximation reasoning of an uncertain and vague concept in a knowledge base. In this paper, we have employed the *Indicator Function* and performed the reasoning for uncertainty, specifically for the rough membership properties of union and intersection. Along with this, we have utilized rough set theory and proposed an approach for an efficient biometric identification. The complexity and efficiency analysis of our proposed approach is also presented in this paper.

Keywords Incomplete information · Biometric identification · Trait extraction Rough sets · Fuzzy sets · Discernible · Uncertainty

1 Introduction

Object classification is one of the rudimentary and challenging problems in artificial intelligence which has been thoroughly studied over past several years. This is the problem of affiliating an object from the set of universe with one out of many

A. Singh (✉) · V. Tiwari · A. N. Tentu
C. R. Rao Advanced Institute of Mathematics, Statistics and Computer Science, University of Hyderabad Campus, Hyderabad, India
e-mail: ajeetsingh.ind@gmail.com

V. Tiwari
e-mail: vikas.tiwari2403@gmail.com

A. N. Tentu
e-mail: naidunit@gmail.com

P. Garg
Indian Institute of Technology Bombay, Mumbai, India
e-mail: priyankagarg112@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_35

predefined existing sets. We, living creatures, normally understand and discern this world around us in terms of objects and its attributes. We know of attributes through perception, and we know the existence of objects through inference. Think that there must be some object upon which we can impose all its attributes. Here, deeply observe that attributes have no independent existence, but only objects can have. For a particular object, if we ignore all of its attributes one by one, what at the end remains is the object; this will be its real and true nature. Objects without the attributes are indescribable [1, 2]. The change and variation are in the attributes, and objects themselves may not change at all. Objects themselves cannot be different from one another, and all the variations observed are only at the level of attributes.

Various approaches like evidence theory (introduced by Glenn Shafer in 1976) and later rough set theory (evolved in 1982, introduced by Zdzislaw Pawlak) have been employed to represent the uncertainty in any expert systems [2–4]. It can be efficiently applied to fuzzy control systems and decision analysis, and it provides a new perspective for the statistical analysis [5, 6]. Image processing, project planning and its scheduling, predictive modeling, association rule mining, knowledge reduction etc., are among its main applications.

We found an application of rough set theory in biometric identification scenario [7] and proposed an approach for an efficient client's biometric identification. In past recent years, several researchers have proposed various schemes [8–12] and algorithmic procedures for biometric identification systems, but most of such schemes lack efficiency which leads to considerably excess time in the process of client's biometric identification.

1.1 Our Contribution

Classification of the universe objects is one of the most embryonic and challenging problems in the domain of artificial intelligence and knowledge discovery. Rough set theory evolved as such a tool which has the power to deal with uncertain, inexact, imprecise, and vague knowledge events in most of the knowledge extraction and machine learning application scenarios. Membership relation is one of the significant and fundamental notions in set theory.

Our main contributions in this paper are as follows:

- We have presented some significant rough membership functional properties. Further, we have employed the *Indicator function* and made an attempt to perform the reasoning for uncertainty, specifically for the rough membership properties of union and intersection.
- We have considered an application scenario and proposed an approach for an efficient biometric identification. The complexity and efficiency analysis of the proposed approach, along with the comparative study, is also given.

1.2 Organization of the Paper

The remaining paper is structured as follows: Sect. 2 contains an overview of rough sets and information systems. Section 3 describes an approximation of classifications. The basic properties of rough membership relations are summarized in Sect. 4. In Sect. 5, we have presented our novel reasoning approach of uncertainty for rough membership relations. In Sect. 6, we have proposed an approach for an efficient biometric identification, and its complexity and efficiency analysis along with the comparative study is also discussed. Finally, conclusions along with our further research directions are given in Sect. 7.

2 Rough Sets and Information Systems

For any set of objects present inside the universe, information systems (ISs) are utilized to represent knowledge. This information system can also be thought of as attribute-value system, data tables, etc. This representation was first originated in 1982 by Pawlak [2] and on the side investigated by numerous researchers.

Rough set acts as a tool to deal with uncertain, inexact, imprecise, and vague knowledge in most of the AI and machine learning application scenarios. It is continuously being intensively investigated, and many practical systems are implemented based on it [13]. Here, this section describes some of the most significant basics related to information systems and rough sets which will be helpful to understand further sections in this paper.

An *Information system* is defined as a pair $A = (U, Attr)$, where U: finite non-empty set in universe; $Attr$: finite and non-empty set of attributes or object features, i.e.,

$$a: \{U \rightarrow v_a\} \forall a \in Attr$$

Here, v_a represents value for a.

In above system, if $Attr$ consists of $C \cup \{d\}$, where C: set of conditional features/attributes and d: decision attributes, then this kind of system is termed as decision system or sometimes decision table. Objects cannot be distinguished from each other if they posses same attribute values. Rough set theory coined this term as *indiscernibility*, and those objects can be termed as *Indiscernible* objects. If for some objects they posses different values, then they are termed as *discernible* objects and this property is *discernibility* property. The M indiscernibility relation can be defined as

$$IND(M) = \{ (p, q) \in U^2 \forall a \in M, a(p) = a(q) \}$$

$[x]_M$ or $[x]_{IND(M)}$ represents *equivalence class* containing an element x.

In an information system, exact representation for some subset of objects in terms of available set of attributes is not possible, and then they can be roughly defined.

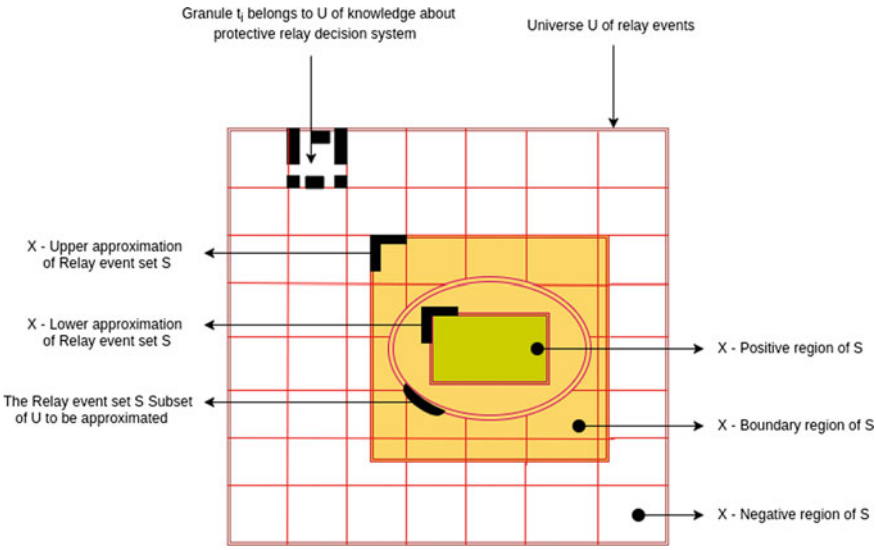


Fig. 1 Concept approximation elements in knowledge space

Therefore, this concept which is uncertain by nature is approximated by *Lower approximation* along with *Upper approximation*.

Let $S = (U, C)$ be an approximation space and X is assumed to be a concept in that space, then lower approximation (LA) can be defined as

$$\underline{C}X = \{x \in U \,|[x] \subseteq X\}$$

and the upper approximation (UA) as

$$\overline{C}X = \{x \in U \,|[x] \cap X \neq \emptyset\}$$

A concept approximation element in universe knowledge space is represented as Fig. 1.

A set X will be *C-definable*, if $\overline{C}X = \underline{C}X$.

Reduct is a significant terminology in rough set theory. In a decision system of form $(U, C \cup \{d\})$, a subset R of conditional attributes C is termed as a reduct if,

$$POS_R(d) = POS_C(d)$$

There is always a possibility of finding more than one reduct in an information system. The intersection of all reducts is termed as *core*, i.e.,

$$CORE(C) = \cap RED(C)$$

3 Classification Through Approximations

This section describes the thought of approximation in classification. The core idea is to perform partition of objects present in universe [1, 3]. These partitioned forms are also termed as granules.

Assume $S = (U, C)$, a representation of an information system. Consider P, Q be the families of subset in universe U such that $Q \subseteq P$ and $|Q| \geq 1$; here, $|Q|$ represents the cardinality of Q. The below set

$$\bigcap_{p \in Q} BN_C(p) \cap \bigcap_{p \in P-Q} (U - BN_C(p))$$

is termed as *Q-boundary region* defined by P and S and can be denoted as $Bd_S(Q, P)$.

The above definition infers a significant proposition that is described as follows:

<u>*Proposition*</u>: Consider $S = (U, C)$ an IS (information system), and suppose P be a family of the pairwise and disjoint subsets of universe set U: $\bigcup P = U$. Suppose $Q \subseteq P$ and $|Q| > 1$. Then, it infers [14].

1. Set $Bd_S(Q, P)$ is definable in S.
2. If $x \in Bd_S(Q, P)$, then $[x]_C \subseteq \bigcup Q$.
3. If $x \in Bd_S(Q, P)$, then for each $p \in P$.
4. $APPROXI_CLASS_{\{S\}}(P)$ - $\{\emptyset\}$ is partition of U.
5. Below equality holds,
   $\underline{C}(\bigcup R) = \bigcup_{p \in R} \underline{C} \times p \cup \bigcup_{|Q|>1; Q \subseteq R} Bd_S(Q, P)$, where $R \subseteq P$

4 Rough Membership Relations: Basic Properties

Membership relation is one of the significant and fundamental notions in set theory. Generally, categories are those items or objects of information that can be expressed through knowledge available. Not all the subsets of the object form categories in available knowledge base. Such subsets will be termed as rough (imprecise by nature), i.e., approximate categories. The only possibility is to define them roughly by utilizing available knowledge. This process is carried out by utilizing two exact categories—*lower approximation* and *upper approximation*. We have listed some significant properties of membership relations with R-lower and R-upper approximations [1, 15, 16].

<u>*Notations*</u>: $x \underline{\in} X$ iff $x \in \underline{R}X$; $x \overline{\in} X$ iff $x \in \overline{R}X$.

Here, $\underline{\in}$ may be interpreted as $\rightsquigarrow x$ surely belongs to X w.r.t. R and $\overline{\in}$ may be interpreted as $\rightsquigarrow x$ possibly belongs to X w.r.t. R.

Properties:

1. "$x \underline{\in} X \Rightarrow x \in X \Rightarrow x \overline{\in} X$.
   Proof: $x \underline{\in} X$ iff $x \in \underline{R}X$ iff $[x] \subseteq X$ but $x \in [x]$ hence, $x \in X$.
   similar way, if $x \in X$, then $[x] \cap X \neq \emptyset$ (since, $x \in [x] \cap X$).
   hence, $x \in \overline{R}X$ iff $x \overline{\in} X$.
   So, $x \underline{\in} X \Rightarrow x \in X \Rightarrow x \overline{\in} X$.
2. $X \subseteq Y \Rightarrow (x \underline{\in} X \Rightarrow x \underline{\in} Y$ and $x \overline{\in} X \Rightarrow x \overline{\in} Y)$.
   Proof: $x \underline{\in} X$ iff $x \in \underline{R}X$ iff $[x] \subseteq X$.
   since, $X \subseteq Y$ means $X \cup Y = Y$ and $X \cap Y = X$.
   hence, $[x] \subseteq Y$ (transitive property).
   $x \in [x]$ iff $x \in \underline{R}Y$ iff $x \underline{\in} Y$.
   now, $x \overline{\in} X$ iff $x \in \overline{R}X$ iff $[x] \cap X \neq \emptyset$.
   since, $X \subseteq Y$, $x \in [x]$, $y \in [y]$, $[x] \cap [y] \neq \emptyset$.
   hence, $[x] \cap Y \neq \emptyset$ iff $x \in \overline{R}Y$ iff $x \overline{\in} Y$.
3. $x \overline{\in} (X \cup Y)$ iff $x \overline{\in} X$ or $x \overline{\in} Y$.
   Proof: $x \overline{\in} (X \cup Y) \Leftrightarrow (x \overline{\in} X) \cup (x \overline{\in} Y)$.
   $x \in \overline{R}(X \cup Y)$ iff $[x] \cap (X \cup Y) \neq \emptyset$ iff $[x] \cap X \bigcup [x] \cap Y \neq \emptyset$ iff $[x] \cap X \neq \emptyset$ $\bigcup [x] \cap Y \neq \emptyset$ iff $(x \in \overline{R}X) \cup (x \in \overline{R}Y)$ iff $(x \overline{\in} X) \cup (x \overline{\in} Y)$.
   So, $x \overline{\in} (X \cup Y) \Leftrightarrow (x \overline{\in} X) \cup (x \overline{\in} Y)$.
4. $x \underline{\in} (X \cap Y)$ iff $x \underline{\in} X$ and $x \underline{\in} Y$.
   Proof: $x \underline{\in} (X \cap Y)$ iff $(x \underline{\in} X) \cap (x \underline{\in} Y)$.
   $x \underline{\in} (X \cap Y)$ iff $[x] \subseteq (X \cap Y)$ iff $x \in \underline{R}(X \cap Y)$ iff $([x] \subseteq X) \cap ([x] \subseteq Y)$ iff $(x \in \underline{R}X) \cap (x \in \underline{R}Y)$ iff $(x \underline{\in} X) \cap (x \underline{\in} Y)$.
   So, $x \underline{\in} (X \cap Y) \Leftrightarrow (x \underline{\in} X) \cap (x \underline{\in} Y)$.
5. $x \underline{\in} X$ or $x \underline{\in} Y \Rightarrow x \underline{\in} (X \cup Y)$.
   Proof: $x \underline{\in} X \cup x \underline{\in} Y \Rightarrow x \underline{\in} (X \cup Y)$.
   $(x \underline{\in} X) \cup (x \underline{\in} Y)$ iff $(x \in \underline{R}X) \cup (x \in \underline{R}Y)$ iff $([x] \subseteq X) \cup ([x] \subseteq Y)$ iff $[x] \subseteq (X \cup Y)$ iff $[x] \subseteq (X \cup Y)$ iff $x \in \underline{R}(X \cup Y)$ iff $x \underline{\in} (X \cup Y)$.
   So, $x \underline{\in} X \cup x \underline{\in} Y \Rightarrow x \underline{\in} (X \cup Y)$.
6. $x \overline{\in} (X \cap Y) \Rightarrow x \overline{\in} X$ and $x \overline{\in} Y$.
   Proof: $x \overline{\in} (X \cap Y)$ iff $x \in \overline{R}(X \cap Y)$ iff $[x] \cap (X \cap Y) \neq \emptyset$ iff $([x] \cap X) \cap ([x] \cap Y) \neq \emptyset$ iff $[x] \cap X \neq \emptyset \bigcap [x] \cap Y \neq \emptyset$ iff $x \in \overline{R}X \cap x \in \overline{R}Y$ iff $x \overline{\in} X$ and $x \overline{\in} Y$.
   So, $x \overline{\in} (X \cap Y) \Rightarrow (x \overline{\in} X) \cap (x \overline{\in} Y)$.
7. $x \underline{\in} (-X)$ iff non $x \overline{\in} X$.
   Proof: $x \underline{\in} (-X)$ iff $x \in \underline{R}(-X)$ iff $[x] \subseteq -X$ iff $[x] \cap X = \emptyset$ iff $x \notin \overline{R}(X)$ iff $x \in -\overline{R}X$ iff $x \overline{\in} (-X)$ iff non $x \overline{\in} X$.
   So, $x \underline{\in} (-X)$ iff non $x \overline{\in} X$.
8. $x \overline{\in} (-X)$ iff non $x \underline{\in} X$.
   Proof: $x \overline{\in} (-X)$ iff $x \in \overline{R}(-X)$ iff $[x] \cap -X \neq \emptyset$ iff $[x] \nsubseteq X$ iff $[x] \nsubseteq X$ iff $x \notin \underline{R}X$ iff $x \in -\underline{R}X$ iff non $x \in \underline{R}X$ iff non $x \underline{\in} X$.
   So, $x \overline{\in} (-X)$ iff non $x \underline{\in} X$."

5 Rough Membership Relations: Novel Reasoning for Uncertainty

In this section, we will consider the rough membership properties of union and intersection. Here, we have employed the *Indicator function* and made an attempt to perform the reasoning for uncertainty.

5.1 Reasoning for Rough Membership Property of Intersection

Consider the Property 4 described in the previous section of this paper,

$$x \underline{\in} (X \cap Y) \text{ iff } x \underline{\in} X \text{ and } x \underline{\in} Y$$

($\star$) *Reasoning through indicator function*:
For any given $x \in U$,
$\delta_{\underline{R}(X \cap Y)}(x) = \delta_{\underline{R}X \cap \underline{R}Y}(x)$;
Here, δ represents indicator function
We have $x \underline{\in} (X \cap Y)$.
$\Rightarrow \delta_{\underline{R}(X \cap Y)}(x)$
$\Rightarrow \underset{y \in [x]_R}{min} (\delta_{X \cap Y}(y))$
$\Rightarrow \underset{y \in [x]_R}{min} (min(\delta_X(y), \delta_Y(y)))$
$\Rightarrow min(\underset{y \in [x]_R}{min} (\delta_X(y)), \underset{y \in [x]_R}{min} (\delta_Y(y)))$
$\Rightarrow min(\delta_{\underline{R}X}(x), \delta_{\underline{R}Y}(x))$
$\Rightarrow \delta_{\underline{R}X \cap \underline{R}Y}(x)$.

5.2 Reasoning for Rough Membership Property of Union

Consider the Property 3 described in the previous section in this paper,

$$x \overline{\in} (X \cup Y) \text{ iff } x \overline{\in} X \text{ or } x \overline{\in} Y$$

($\star\star$) Reasoning through indicator function:
For any given $x \in U$,
$\delta_{\overline{R}(X \cup Y)}(x) = \delta_{\overline{R}X \cup \overline{R}Y}(x)$;
Here, δ represents indicator function
We have $x \overline{\in} (X \cup Y)$.
$\Rightarrow \delta_{\overline{R}(X \cup Y)}(x)$
$\Rightarrow \underset{y \in [x]_R}{max} (\delta_{X \cup Y}(y))$
$\Rightarrow \underset{y \in [x]_R}{max} (max(\delta_X(y), \delta_Y(y)))$

$\Rightarrow max(\underset{y\in[x]_R}{max}(\delta_X(y)), \underset{y\in[x]_R}{max}(\delta_Y(y)))$
$\Rightarrow max(\delta_{\overline{R}X}(x), \delta_{\overline{R}Y}(x))$
$\Rightarrow \delta_{\overline{R}X\cup\overline{R}Y}(x)$.

So, through above reasonings ($\star$) and ($\star\star$), we have shown that how indicator function can be employed in reasoning for some rough membership functional properties in a novel way.

6 Rough Set-Based Approach for a Computationally Efficient Biometric Identification: An Application Scenario

We have considered and modeled an application of rough sets in biometric identification scenario and proposed an approach for an efficient clients' biometric identification. In the practical scenario of adopting biometric identification systems, efficiency is considered as main obstacle; i.e., the identification process must be performed in a reasonable time. Here, we have proposed an approach for an efficient biometric identification. Our approach utilizes rough set phenomenon which achieves better efficiency in biometric identification process.

6.1 Proposed System Model and Design Goal

As the reinforcement of efficient and computationally intelligent methodologies for biometric sensors is maturing rapidly, a logically efficient client identification schema utilizing biometric traits is the demand of hour across varying applications. A client's fingerprint trait database consists of huge information in the form of FingerCode vectors.

The workflow is represented in Fig. 2. The main entity in our system model is the client which is having his/her biometric finger traits. This trait/feature collection vector is termed as FingerCode. FingerCode is chain which consists of N-independent trait codes, which are 8-bit integers typically. This entire collection can be thought

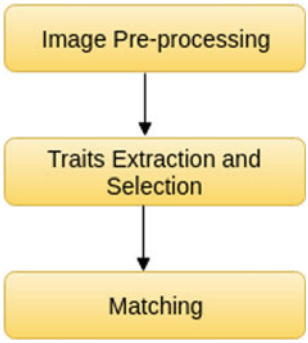


Fig. 2 Workflow in RS-based BI model

of as an information system. Our procedure will extract and select the minimal number of trait/feature subset which can exhibit the same knowledge system as it was exemplified by the initial traits.

6.2 The Proposed Approach

For set of objects present inside the universe, information systems (ISs) are utilized to represent knowledge. This IS can also be thought of as feature-value system table. The representation of IS in our proposed procedure is as given below:

An information system is a pair $IS = \{FC, T\}$ where

$FC\{FC_1, FC_2, \ldots, FC_n\}$: non-empty and finite set of each individual FingerCode.

$T\{T_1, T_2, \ldots, T_n\}$: non-empty and finite set of traits/attributes.

The steps involved in our proposed procedure are as below:

Algorithm 1 RS-based Approach for an Efficient Biometric Identification-

1: Input: In the form of { FingerCode, trait-value } information table.
2: *Begin procedure*
3: Construct Discernibility matrix for above "FingerCode, trait-value" information table.
Note: *A discernibility matrix of* $IS = (FC, T)$ *is a symmetric* $|FC| X |FC|$ *matrix where entries are as-*
$c_{ij} = \{t \in T | t(x_i) \neq t(x_j)\}\ i, j = 1, \ldots, |FC|$
Each c_{ij} *consists of those traits that discern among objects* i & j.
4: For each row of discernibility matrix, computation of discernibility function will be as follows -
$f_1, f_2, ..., f_{|FC|} = \bigwedge\{\bigvee c_{ij} | 1 \leq j \leq i \leq |FC|, c_{ij} \neq \emptyset\}$
5: Evaluate resultant function as -
$F_{RES} = \bigcap\{f_i, 1 \leq i \leq |FC|\}$
6: *Begin traits optimization sub-procedure* {
7: while (($F_{RES}, \{f_i, 1 \leq i \leq |FC|\}$) are not already in minimized form)
8: Apply Absorption Rule to remove the super-sets in ($F_{RES}, \{f_i, 1 \leq i \leq |FC|\}$)
9: replace strongly shrinkable traits ($F_{RES}, \{f_i, 1 \leq i \leq |FC|\}$)
10: $fre \Leftarrow$ most_frequent attribute ($F_{RES}, \{f_i, 1 \leq i \leq |FC|\}$)
11: Expansion Rule applied ($fre, (F_{RES}, \{f_i, 1 \leq i \leq |FC|\})$)
12: substitute extreme shrinkable classes ($F_{RES}, \{f_i, 1 \leq i \leq |FC|\}$)
13: $\Re \Leftarrow$ calculate minimal FingerCode traits vector(reduct) ($F_{RES}, \{f_i, 1 \leq i \leq |FC|\}$)
14: returns $\Re$
15: } *End traits optimization sub-procedure*
16: Set of all prime implicants(in F_{RES}) are obtained. These can also termed as 'Reduct-set($\Re$-set)'.
17: *End procedure*

This way each client can be uniquely identified in the biometric identification process with the minimal FingerCode trait vector subset extracted above.

6.3 Complexity and Efficiency

This section presents the complexity and efficiency analysis for the proposed approach.

Table 1 Comparison and summary

Comparison with existing algorithms

Algorithm	Orientation methodology	Ridge trait matching	Observation
Jain et al. [11]	Gabor filters	Euclidean distance	Much complex behavior for large database
Hill [17]	Pole–zero–gain model	Discriminatory skeleton	Discriminate and incomplete fingerprint obtained
Ross et al. [18]	Minutiae triplicates	Linear vector integral convolution, streaklines	Fingerprint partially reconstructed
Cappelli et al. [19]	Revamped pole–zero–gain model	Gabor filters	Fallacious minutiae obtained in fingerprint reconstruction
Huang et al. [10]	Backtracking protocol	Garbled circuit evaluation	An efficient oblivious information retrieval, privacy preserving fingerprints matching
Feng and Jain [20]	Octonion structured short distance minutiae	AM-FM filter model	Fallacious minutiae and blocking effect
Li and Kot [8]	Octonion structured short distance minutiae	AM-FM filter model	In continuous phase ridge frequency changes
Cao and Anil [9]	Orientation periodic dictionary	Continuous periodic patch dictionary	Involves orientation field learning and pattern matching
Hahn and Hur [12]	Symmetric dictionary encryption	Private key-based homomorphic computation	An efficient fingerprint identification, involves privacy preservation scenario in cloud
Proposed procedure	Information system orientation, approximation and uncertainty logic theory	Rough set-theoretic methodology, partial membership of sets	Efficiently finds minimal length finger-trait subset for matching, does not require any probability or ridge membership degree measure

Constructing $|FC| \times |FC|$ symmetric matrix individually involves $|n|$ comparisons. $|n|$ represents total number of FingerCodes in information system database. Therefore, total number of comparisons are bounded as $O(|n| \times |n|)$.

Trait shrinking phase requires $O(a \cdot c + a^2)$ complexity, where a denotes number of traits and c denotes number of clauses. This way, the biometric identification process is more efficient and can be performed in reasonable time with the reduced FingerCode trait vector subset.

6.4 Comparative Analysis

Here, we have presented the comparative analysis of our proposed procedure with some significant state-of-the-art approaches developed in past years. The representation is given in Table 1.

7 Conclusions

This section concludes the paper in terms of an overview of our contribution and our further research work.

7.1 Contribution

With granulated knowledge scenarios, we specifically handle any imprecise concept with the approximations represented by the subsets of universe knowledge base which are in the form of granules. These structures have been successfully employed in rough sets as well as neighborhood systems.

This paper conveys some significant rough membership functional properties along with their analytical behavior. Also, we have employed the *Indicator function* and made an attempt to perform the reasoning for uncertainty, specifically for the rough membership properties of $\cup$ and $\cap$. Later, we have considered an application scenario and proposed an approach for an efficient biometric identification along with its complexity and an efficient comparative analysis.

7.2 Further Research Work

Further, we would like to explore more about the functional behavior of an uncertain concept in the universe and will make an attempt to deeply analyze the rough membership relations. Investigations toward this direction may produce some practically useful and interesting results.

References

1. Pawlak, Z.: Rough Sets: theoretical aspects of reasoning about data. Kluwer Academic Publishing, Dordrecht (1991)
2. Pawlak, Z.: Rough sets. Int. J. Comput. Inf. Sci. **11**, 341–356 (1982)
3. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
4. Skowron, A.: The Rough Set Theory as a Basis for the Evidence Theory. ICS Research Report 2/91, 53 pp
5. Zadeh, L.A.: Fuzzy sets. Inform. Control **8**, 338–353 (1965)
6. D. Dubos and H. Prade. Fuzzy Sets and Systems: Theory and Applications, Academic Press, (1980)
7. Bolle, R., Pankanti, S.: Biometrics: personal identification in networked society. Kluwer Academic Publishers (1998)
8. Li, S., Kot, A.C.: An improved scheme for full fingerprint reconstruction. IEEE Trans. Inf. Forensics Secur. (2012)
9. Cao, K., Anil, J.: Learning fingerprint reconstruction: from minutiae to image. IEEE Trans. Inf. Forensics Secur. (2015)
10. Huang, Y., Malka, L., Evans, D., Katz, J.: Efficient privacy-preserving biometric identification. In: NDSS (2011)
11. Jain, A.K., Prabhakar, S., Hong, L., Pankanti, S.: Filterbank-based fingerprint matching. IEEE Trans. Image Process. (2000)
12. Hahn, C., Hur, J.: Efficient and privacy-preserving biometric identification in cloud. ICT Express **2**, 135–139 (2016)
13. Skowron, A., Grzymala-Busse, J.: From the Rough Set Theory to the Evidence Theory. ICS Research Report 8/91, 49 pp
14. Pawlak, Z., Skowron, A.: Rough membership functions: a tool for reasoning with uncertainty. In: Algebraic Methods in Logic and in Computer Science, vol. 28. Banach Center Publications, Institute of Mathematics, Polish Academy of Sciences, Warszawa (1993)
15. Pawlak, Z., Skowron, A.: Rough membership functions, ICS Research Report 10/91, Warsaw University of Technology
16. Scott, D.: Domains for denotational semantics, a corrected and expanded version of a paper presented at ICALP 82. Aarhus, Denmark (1982)
17. Hill, C.J.: Risk of masquerade arising from the storage of biometrics. B.S. thesis, The Department of Computer Science, Australian National University, Canberra, ACT, Australia (2001)
18. Ross, A., Shah, J., Jain, A.K.: From template to image: reconstructing fingerprints from minutiae points. IEEE Trans. Pattern Anal. Mach. Intell. **29**(4), 544–560 (2007)
19. Cappelli, R., Maio, D., Lumini, A., Maltoni, D.: Fingerprint image reconstruction from standard templates. IEEE Trans. Pattern Anal. Mach. Intell. **29**(9), 1489–1503 (2007)
20. Feng, J., Jain, A.K.: Fingerprint reconstruction: from minutiae to phase. IEEE Trans. Pattern Anal. Mach. Intell. **33**(2), 209–223 (2011)

Butterfly Constrained Optimizer for Constrained Optimization Problems

Abhishek Kumar, Tarun Maini, Rakesh Kumar Misra and Devender Singh

Abstract An extension of the new optimization algorithm, butterfly optimizer (BO) for the constrained optimization problem is discussed in this paper. This version of BO is called butterfly constrained optimizer (BCO) which mimics the mate-locating behaviors of male butterfly and his behavior toward sunspots. In BCO, the location of male butterflies presents the trial solutions, and sunspots and dark-spots represent the feasible and infeasible region of search space. Two major mate-locating behaviors, patrolling and perching, are applied to generate new location of butterflies toward the feasible reason of search space to optimize the problem without violating any constraints. In this paper, five benchmark constrained optimization problems are considered to analyze the performance of BCO, and the benchmark results are compared with well-known state-of-the-art constrained techniques. Comparative result shows that the performance of BCO concerning its optimization capability, efficiency, and accuracy is better than other.

Keywords Butterfly optimizer · Butterfly constrained optimizer · Constrained optimization problem

A. Kumar (✉) · T. Maini · R. Kumar Misra · D. Singh
Department of Electrical Engineering, IIT (BHU) Varanasi, Varanasi 221005, Uttar Pradesh, India
e-mail: abhishek.kumar.eee13@iitbhu.ac.in

T. Maini
e-mail: tmaini.rs.eee13@iitbhu.ac.in

R. Kumar Misra
e-mail: rkmishra.eee@iitbhu.ac.in

D. Singh
e-mail: dsingh.eee@iitbhu.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_36

1 Introduction

Nowadays, to compute the solution of different optimization issues, nature-inspired techniques (NITs) are extensively used. However, most of the NITs are mainly designed for the unconstrained search space. Although, a lot of constrained optimization problems are originated from many field of engineering and sciences. A nonlinear standard constrained optimization problem can be formulated as:

$$\text{Minimize } f(\mathrm{x}), \mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n) \epsilon R^n$$

where $f(\mathrm{x})$ is the objective or cost function and $(x \epsilon S \cup S')$.

Here, S is a $n-$dimension hypercube space in R^n bounded by the bound constraints as:

$$LB(i) \leq \mathrm{x}_i \leq UB(i), \text{ where } 1 \leq i \leq n$$

S' is the feasible region and formed by a set of m nonlinear or linear equality or inequality constraints ($m \geq 0$):

$$S' = \{x \epsilon R^n | h_k(x) = 0, g_i(x) \leq 0, 1 \leq i \leq a \leq j \leq m\}$$

where a and $(m - a)$ are the total number of inequality and equality constraints, respectively.

Any inequality constraints, which satisfies the $g_j(\bar{x}) = 0$, where $\bar{x}$ is the any point in region F, then that inequality constraint is active at $\bar{x}$.

Among these NITs, mostly inspired by swarming and collective foraging behaviors of animals and bio-organisms, are called swarm intelligence-based optimization algorithms (SIs). Some popular SIs are: artificial bee colony [1] (ABC), ant colony optimization [2] (ACO), and particle swarm optimization [3] (PSO), etc. Bat algorithm [4] (BA) and gray wolf optimizer [5] (GWO) are new SIs. Collective behaviors of butterflies also inspired to design optimization methods based on the intelligence of butterflies. Butterfly optimizer (BO), monarch butterfly optimization [6] (MBO), and butterfly particle swarm optimization [7] (BFPSO) are three newly developed optimization techniques based on intelligent behaviors of butterflies. BFPSO is inspired by the nectar foraging actions of butterflies [7]. MBO is developed by modeling the butterflies' migration behavior [6].

It is debatably said that locating a feasible solution on search space is more important than the minimizing the cost function (objective function) in real-life constrained optimization problems. The main dispute of constrained optimization is to balance the searching between infeasible and feasible region. The most popular way of balancing is to incorporate constraint handling techniques in optimization technique. Numerous are proposed to handle the constraints, popular constraint handling techniques (CHT) are repair method, special representations, penalty function method, and operators, multi-objective method, etc.

Deb suggests three comparison rules for ranking the solution according to feasibility an objective function value for constrained optimization [8]:

1. All infeasible solutions are worse than to all feasible ones [8].
2. The feasible solution having better objective function value is considered better than other feasible solution [8].
3. The infeasible solution having better constraint violation is considered better than other infeasible solution [8].

Butterfly optimizer (BO) [9] is a simple NITs and is an efficient approach to global unconstrained optimization. In this paper, BO is incorporated with Debs criteria to handle the constraints in case of constrained optimization problems. This constrained version of BO is called as butterfly constrained optimizer (BCO). BCO is compared with other state-of-the-art methods on five common test problems. From the experimental analysis, BCO can effectively solve the constrained optimization problems.

2 Butterfly Constrained Optimizer

Butterfly optimizer (BO) is a newly developed unconstrained optimization technique which is motivated by the mate-locating behaviors: perching and patrolling behavior of male butterflies. An extension of BO, butterfly constrained optimizer (BCO), is discussed in this section, which is a dual-population-based algorithm and capable of solving constrained optimization problems.

Male butterflies always pursue female butterflies and by using mate-locating behaviors: perching and patrolling they locate the position of female butterflies. Some essential property of male butterflies is mimicked in BCO. These properties are as follows:

1. Butterflies can memorize any location using different sign of surrounding.
2. Butterflies can fly with fixed speed continuously in same direction.
3. Butterflies can remember the direction of their flying.
4. Male butterflies always prefer sunspots or higher temperature locations in place of dark-spots to sit.

In this algorithm, two type of population is initialized randomly on the search space. The first population contains the location of auxiliary butterflies, and other population represents the position of main butterflies. In BCO, the first population updates their values to improve the constraint violation or feasibility in every iteration. On the other hand, second population updates to improve the constraint violation as well as their cost values in every iteration.

If vector $\bar{x}_i$ is the location of ith main butterfly and vector $\bar{y}_i$ is the location of ith auxiliary butterfly, then set of all vectors $\bar{x}$ and $\bar{y}$ represents the both the population of BCO. Let us create the matrix X and Y of both populations which contain the present position vector of all main butterflies and auxiliary butterflies, respectively, and defined by the Eqs. (1) and (2).

$$X = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ :: \\ \bar{x}x_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & :: & x_{1D} \\ x_{21} & x_{22} & :: & x_{2D} \\ :: & :: & :: & :: \\ x_{N1} & x_{N2} & :: & x_{ND} \end{bmatrix} \tag{1}$$

$$Y = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ :: \\ \bar{y}_N \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & :: & y_{1D} \\ y_{21} & y_{22} & :: & y_{2D} \\ :: & :: & :: & :: \\ y_{N1} & y_{N2} & :: & y_{ND} \end{bmatrix} \tag{2}$$

where D is the size of search space and N is the total number of male butterflies.

2.1 Modeling of Perching Behavior

In perching, all auxiliary butterfly initiates flying and appears near to their respective main butterfly. When main butterfly senses any fluttering body near to him, it starts to follow that object. So, ith main butterfly starts following the ith auxiliary butterfly. Auxiliary butterflies sit on a new position $\bar{y}_i$ which is calculated as follows:

1. Vectors $\bar{p}$, $\bar{q}$, and $\bar{r}$ of size $NX1$ are filled with integers between 1 to N in such a way that elements of these vectors p_i, q_i, r_i, respectively, and i are always different integers to each other(i.e., $p_i \neq q_i \neq r_i \neq i$).
2. Elements of vector $\bar{p}$ are also different to each other.
3. A random direction represented by a vector m is selected where $m_i \epsilon [1, D]$.
4. Then, a trial of new position of auxiliary butterflies $\bar{t}_i$ is calculated by using Eqs. (3)–(5).
5. If the constraint violation of $\bar{t}_i$ is better or equal to the constraint violation of $\bar{y}_i$, then $\bar{t}_i$ replaces the $\bar{y}_i$ for the next iteration, as shown in Eq. (6).

$$t_{im_i}^{k+1} = \Re(x_{p_i m}^k, y_{p_i m}^k) + F * (\Re(x_{q_i m}^k, y_{q_i m}^k) - \Re(x_{r_i m}^k, y_{r_i m}^k)) \tag{3}$$

$$t_{ij}^{k+1} = x_{ij}^k, \text{ where } j \neq m_i \tag{4}$$

$$\bar{t}_i = [t_{i1}, t_{i2}, \ldots, t_{tD}]' \tag{5}$$

$$\bar{y}_i^{k+1} = \begin{cases} \bar{t}_i^k, & \text{if } \phi(\bar{t}_i^k) \leq \phi(\bar{y}_i^k) \\ \bar{y}_i^k, & \text{otherwise.} \end{cases} \tag{6}$$

where $\Re(a, b)$ is random selection operator, where probability of selection of a and b is equal.

The position of main butterflies is also updated at the end of every iteration. The main butterflies fly along their auxiliary butterfly, and both reach to new position. If the objective function of new position of ith auxiliary butterfly $f_{\bar{y}_i}$ is less than objective function value of ith main butterfly $f_{\bar{x}_i}$, $\bar{y}_i$ replaces the $\bar{x}_i$ in second population for the next iteration. If $\bar{x}_i$ is not updated for the next iteration, then it goes to patrolling phase. The equation used to update the position of the main butterfly is given by:

$$\bar{x}_i^{k+1} = \begin{cases} \bar{y}_i^{k+1}, \text{ if } f_{\bar{y}_i{}^{k+1}} \leq f_{\bar{x}_i^k} \\ \bar{x}_i^k, \text{otherwise} \end{cases} \tag{7}$$

2.2 Modeling of Patrolling Behavior

The main butterfly continues flying in patrolling. The starting flying direction is always in the direction of previous patrolling direction. After a certain portion of the time, d, the main butterfly starts following the most attractive object and switches its direction. After following, the most attractive object stops flying and reaches to a new position $\bar{u}_i$. Mathematically, this new position is computed by Eq. (8).

$$\bar{u}_j^{k+1} = \bar{x}_j^k + d * \bar{v}_j^k + (\bar{x}_{maxuv}^k - \bar{x_j}^k) * \text{rand}(-1, 1) \tag{8}$$

Constraint violation and objective function values at $\bar{u}_j^{k+1}$, $f_{\bar{u}_j^{k+1}}$ and $\phi\bar{u}_j^{k+1}$ with the constraint violation and objective function values at $\bar{x}_j^{k+1}$, $f_{\bar{x}_j^{k+1}}$ and $\phi(\bar{x}_j^{k+1})$ according to Debs rule.
i.e.,

$$\bar{x_j}^{k+1} = \begin{cases} \bar{u}_j^{k+1}, \text{ if } \phi(\bar{u}_j^{k+1}) < \phi(\bar{x}_j^{k+1}) \\ \bar{u}_j^{k+1}, \text{elseif } \phi(\bar{u}_j^{k+1}) == \phi(\bar{x}_j^{k+1}) \\ \qquad f_{\bar{u_j}{}^{k+1}} \leq f_{\bar{x}_j^{k+1}} \\ \bar{x}_j^k, \quad \text{else} \end{cases} \tag{9}$$

Velocity is updated as given below:

$$\bar{v}_j^{k+1} = \begin{cases} \bar{u}_j^{k+1} - \bar{x}_j^k, \text{ if } f_{\bar{u}_j^{k+1}} \leq f_{\bar{x}_j^{k+1}} \\ d * \bar{v}_j^k, \text{ else} \end{cases} \tag{10}$$

There is a chance that any butterfly might go out of the limits of search space; this is called bound violation. To control this issue, butterfly halts at the boundary of the search space if it violates the bound constraints.

Table 1 Standard deviation and mean of minimum value, get after maximum number of function evaluation by BCO, and feasibility rate (FR), success rate (SR), and mean of total number of function evaluation taken by BCO to reach desired error (0.00001) (mNFEs) for 25 independent runs, for constrained benchmark problems $tp1$ to $tp5$

Prob	Mean	stdev	FR	SR	mNFEs
tp1	−1.50E+01	0	100	100	26489
tp2	−6.96E+03	1.64E−12	100	100	34357
tp3	−9.58E−02	0	100	100	2049
tp4	7.50E−01	3.91E−08	100	100	34630
tp5	0	0	100	100	138

3 Experimental Results and Discussion

The functioning of BCO is examined on five well-known constrained problems, and benchmark result of BCO is compared with the result of other popular constrained variants of PSO, viz. basic PSO [10], DMSPSO [11], CSHPSO [12], SHPSO [13], and DE, viz. basic DE [14], jDE-2 [15], COMDE [16], DSSMDE [17], PCX [18], PESO+ [19].

These five constrained benchmark problems are selected from [20] and [21]. The properties and characteristics of constrained benchmark problems are explained in the appendix.

The benchmark experiment was carried out on an Intel Core-i5 4440 3.10 GHz CPU, 10GB RAM, Windows 10 64 bits OS, with MATLAB 2016b. In this experiment, maximum number of function evaluations (FEs) is allowed to $2000 * d$, where d is the size of search space. The benchmark results of all the other algorithms were considered from their referred paper. For the fair comparison, the experimental procedure is used same as described in other papers. A total number of the independent run is fixed to 30 for each of benchmark problems. The standard deviation and mean of the best value found in all 25 runs for all methods are taken (Table 1).

Table 2 shows the success performance of the BCO, Basic PSO, DMS-PSO, CSH-PSO, SHPSO, DE, jDE-2, COMDE, PCX, and PESO+ for benchmark problems $tp1$ to $tp5$. From Table 2, it is confirmed from Table 2 that the success performance of BCO is best for test problems $tp1$ and $tp5$. However, in the case of test problems $tp2$, $tp3$, and $tp4$, success performance of COMDE, DE, and DMS-PSO, respectively, is best as compared to other algorithms.

For better comparison, Friedman rank test has been done based on the SP of all the algorithms. The result of rank test is reported in the last row of Table 2. As shown in Table 2, the rank of BCO is the best among the other algorithm. Thus, this test suggests that the solving capability of BCO is the better than the other algorithms.

Table 2 Success performance of BCO, basic PSO, DMSPSO, CSHPSO, SHPSO, DE, jDE-2, COMDE, PCX, and PESO+ for benchmark problems $tp1$ to $tp5$ and Fridman ranking(FR) of BCO, Basic PSO, DMS-PSO, CSHPSO, SHPSO, DE, jDE-2, COMDE, PCX, and PESO+ based on Success Performance

Algorithm	*tp*1	*tp*2	*tp*3	*tp*4	*tp*5	*FRT*
BCO	**26489**	34357	2049	34630	**138**	**2.4**
Basic PSO	122166	82713	50496	66528	61128	9.8
DMS-PSO	33332	27635	4124	**14625**	5408	3
CSHPSO	49120	51991	49088	48903	200080	6.8
SHPSO	50250	53120	50218	50032	201194	8
DE	33770	53364	**1832**	50276	7330	5
jDE-2	50386	29488	3236	53928	6355	5.4
COMDE	130000	**12000**	4000	50000	6000	5
DSS-MDE	144953	19076	3101	498553	30615	7.2
PCX	55204	33821	2825	38688	8960	5
PESO+	101532	56507	6124	450099	8088	8.4

4 Conclusion

For solving continuous constrained optimization problems, BCO has been proposed in this paper. The results of BCO are compared with the results of popular constrained variants of PSO and DE. The comparative study suggests that the BCO outperformed others methods regarding success performance for the most of the problems. Friedman rank test shows that BCO provides the superior performance as compared to other methods on the constrained optimization problems.

In future, the behavior of BCO is analyzed for real-life optimization problems and more number of complex constrained problems containing small feasible regions.

Appendix

Test Problem-1 (tp1)

Minimize [20],

$$f(\mathbf{y}) = 5\sum_{i=1}^{4} y_i - 5\sum_{i=1}^{4} y_i^2 - \sum_{i=5}^{13} y_i$$

Subject to,

$$
\begin{aligned}
g_1(\mathbf{y}) &= 2y_1 + 2y_2 + y_{10} + y_{11} - 10 \leq 0 \\
g_2(\mathbf{y}) &= 2y_1 + 2y_3 + y_{10} + y_{12} - 10 \leq 0 \\
g_3(\mathbf{y}) &= 2 * y_2 + 2y_3 + y_{11} + y_{12} - 10 \leq 0 \\
g_4(\mathbf{y}) &= -8y_1 + y_{10} \leq 0 \\
g_5(\mathbf{y}) &= -8y_2 + y_{11} \leq 0 \\
g_6(\mathbf{y}) &= -8y_3 + y_{12} \leq 0 \\
g_7(\mathbf{y}) &= -2y_4 - y_5 + y_{10} \leq 0 \\
g_8(\mathbf{y}) &= -2y_6 - y_7 + y_{11} \leq 0
\end{aligned}
$$

$$0 \leq y_i \leq 1, i = 1, \ldots, 9$$

$$0 \leq y_i \leq 100, i = 10, 11, 12$$

$$f(\mathbf{y}^*) = -15.$$

Test problem-2 (tp2)

Minimize [21],

$$f(\mathbf{y}) = (y_1 - 10)^3 + (y_2 - 20)^3$$

Subject to,

$$
\begin{aligned}
g_1(\mathbf{y}) &= -(y_1 - 5)^2 - (y_2 - 5)^2 + 100 \leq 0 \\
g_2(\mathbf{y}) &= (y_1 - 6)^2 + (y_2 - 5)^2 - 82.81 \leq 0
\end{aligned}
$$

$$13 \leq y_1 \leq 100, 0 \leq y_2 \leq 100$$

$$f(\mathbf{y}^*) = -6961.81387558015.$$

Test problem-3 (tp3)

Minimize [21],

$$f(\mathbf{y}) = \frac{-sin^3(2\pi y_1)sin(2\pi y_2)}{y_1^3(y_1 + y_2)}$$

Subject to,

$$g_1(\mathbf{y}) = y_1^2 - y_2 + 1 \leq 0$$
$$g_2(\mathbf{y}) = 1 - y_1 + (y_2 - 4)^2 \leq 0$$

$$0 \leq y_1 \leq 10, 0 \leq y_2 \leq 10$$

$$f(\mathbf{y}^*) = -0.0958259414180359.$$

Test Problem-4 (tp4)

Minimize [21],

$$f(\mathbf{y}) = y_1^2 + (y_2 - 1)^2$$

Subject to,

$$h(\mathbf{y}) = y_2 - y_1^2 = 0$$

$$-1 \leq y_1 \leq 1, -1 \leq y_2 \leq 1$$

$$f(\mathbf{y}^*) = 0.7499.$$

Test Problem-5 (tp5)

Minimize [21],

$$f(\mathbf{y}) = \frac{-100(100 - (y_1 - 5)^2 - (y_2 - 5)^2 - (y_3 - 5)^2)}{100}$$

Subject to,

$$g(\mathbf{y}) = (y_1 - p)^2 + (y_2 - q)^2 + (y_3 - r)^2 - 0.0625 \leq 0$$

$$0 \leq y_i \leq 10, (p, q, r) = 1, 2, \ldots, 9$$

$$f(\mathbf{y}^*) = -1.$$

References

1. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)
2. Dorigo, M., Birattari, M.: Ant colony optimization. In: Encyclopedia of Machine Learning, pp. 36–39. Springer (2010)
3. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of Machine Learning, pp. 760–766. Springer (2010)
4. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO), pp. 65–74. Springer (2010)
5. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
6. Wang, G.-G., Deb, S., Cui, Z.: Monarch butterfly optimization. Neural Comput. Appl. 1–20 (2015)
7. Bohre, A., Agnihotri, G., Dubey, M.: Hybrid butterfly based particle swarm optimization for optimization problems. In: 2014 First International Conference on Networks Soft Computing (ICNSC), pp. 172–177, Aug 2014
8. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. **186**(2), 311–338 (2000)
9. Kumar, A., Misra, R.K., Singh, D.: Butterfly optimizer. In: 2015 IEEE Workshop on Computational Intelligence: theories, applications and future directions (WCI), pp. 1–6, Dec 2015
10. Eberhart, R.C. Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, vol. 1, pp. 39–43. New York, NY (1995)
11. Liang, J.J., Suganthan, P.N.: Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism. In: IEEE Congress on Evolutionary Computation, (CEC), pp. 9–16. IEEE (2006)
12. Yadav, A., Deep, K.: An efficient co-swarm particle swarm optimization for non-linear constrained optimization. J. Comput. Sci. **5**(2), 258–268 (2014)
13. Yadav, A., Deep, K.: Shrinking hypersphere based trajectory of particles in PSO. Appl. Math. Comput. **220**, 246–267 (2013)
14. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
15. Huang, V.L., Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for constrained real-parameter optimization. In: IEEE Congress on Evolutionary Computation, (CEC), pp. 17–24. IEEE (2006)
16. Mohamed, A.W., Sabry, H.Z.: Constrained optimization based on modified differential evolution algorithm. Inf. Sci. **194**, 171–208 (2012)
17. Zhang, M., Luo, W., Wang, X.: Differential evolution with dynamic stochastic selection for constrained optimization. Inf. Sci. **178**(15), 3043–3074 (2008)
18. Sinha, A., Srinivasan, A., Deb, K.: A population-based, parent centric procedure for constrained real-parameter optimization. In: IEEE Congress on Evolutionary Computation, (CEC), pp. 239–245. IEEE (2006)
19. Munoz-Zavala, A.E., Hernandez-Aguirre, A., Villa-Diharce, E.R., Botello-Rionda, S.: Peso + for constrained optimization. In: IEEE Congress on Evolutionary Computation, (CEC), pp. 231–238. IEEE (2006)
20. Floudas, C.A., Pardalos, P.M.: A Collection of Test Problems for Constrained Global Optimization Algorithms, vol. 455. Springer Science & Business Media (1990)
21. Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol. Comput. **7**(1), 19–44 (1999)

Multimodal Medical Image Fusion Based on Fuzzy Sets with Orthogonal Teaching–Learning-Based Optimization

T. Tirupal, B. Chandra Mohan and S. Srinivas Kumar

Abstract The purpose of an image fusion for medical images is to associate a number of images gained from many bases to a solitary image appropriate for better analysis. The vast majority of the best in class image fusing systems are based on non-fuzzy sets, and the fused image so obtained lags with complementary information. Fuzzy sets are strong-minded to be more appropriate for medical image processing as more hesitations are considered compared with non-fuzzy sets. In this paper, a procedure for efficiently fusing multimodal medical images is presented. In the proposed method, images are initially converted into intuitionistic fuzzy images (IFIs), and another target work called intuitionistic fuzzy entropy (IFE) is utilized for membership and non-membership capacities to accomplish the finest estimation of the bound. Next, the IFIs are compared using the fitness function, entropy. Then, orthogonal teaching–learning-based optimization (OTLBO) is introduced to optimize combination factors that change under teaching phase, and learner phase of OTLBO. Finally, the fused image is achieved using optimal coefficients. Simulations on several pairs of multimodal medical images are performed and matched with the current fusion approaches. The dominance of the proposed technique is presented and justified. Fused image quality is also verified with various quality metrics, such as peak signal-to-noise ratio (PSNR), universal quality index (UQI), structural similarity (SSIM), correlation coefficient (CC), entropy (E), spatial frequency (SF), edge information preservation ($Q^{AB/F}$), and standard deviation (SD).

Keywords Image fusion · Diagnosis · Fuzzy sets · IFI · OTLBO
Spatial frequency

T. Tirupal (✉) · S. Srinivas Kumar
JNTUK, Kakinada, India
e-mail: tirutalari@gmail.com

S. Srinivas Kumar
e-mail: samay_ssk2@yahoo.com

B. Chandra Mohan
Bapatla, India
e-mail: chandrabhuma@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_37

1 Introduction

Multimodal medical image fusion (MMIF) [1] is the progression of bonding two multimodal medical images to increase the quality of output image. In merging medical image pairs, such as computed tomography–magnetic resonance imaging (CT–MRI), magnetic resonance imaging–magnetic resonance angiography (MRI–MRA), X-ray–vibro-acoustography (X-ray–VA) [2], MRI–PET (magnetic resonance imaging—positron emission tomography) [3], and MRI–SPECT (magnetic resonance imaging–single photon emission computed tomography), image fusion is gradually achieving consequence in healthcare and modern medicine. These sets of images are used for mining clinical facts that are complimentary in nature. For example, CT image ambiances with a lesser amount of distortion and conveys details regarding condensed structures like bones. MRI offers neurotic soft fleshy tissue evidence and MRA intellects easily brain defects. X-ray identifies fractures and irregularities in bone position while VA provides depth and width of a disease object. PET and SPECT provide functional and metabolic information of the human brain. Hence, all significant data cannot be obtained in a single image and henceforth MMIF is essentially necessary to achieve all probable comprehensive data in a single combined image called fused image.

Fuzzy sets play a substantial role in image handling to reduce the obscurity and ambiguity existing in images. Ambiguity in fuzzy sets is taken as a membership function which deceits the interim [0, 1] where 0 characterizes no membership and 1 characterizes full membership. Fuzzy set theory anticipated by Zadeh [4] in 1965 is not able to directly model uncertainties. Atanassov [5] in 1986 presented an intuitionistic fuzzy set (IFS), a general form of fuzzy set hypothesis. IFS takes two ambiguity parameters—the membership degree and non-membership degree (due to the hesitation degree). Real-time complications with indefinite understanding are resolved effectively by the non-membership degree. In each period of image handling, numerous vulnerabilities happen. These vulnerabilities are reduced by means of IFSs, and the image is boosted in terms of contrast of the image.

Rao et al. [6] formerly advanced a universal optimization process called teaching–learning-based optimization (TLBO). The TLBO shows nearly common features of former evolutionary computation (EC) procedures which are a population-based iterative learning process. In any case, TLBO analyzes for an optimum over every learner attempting to accomplish the understanding of the teacher. TLBO attains optimum outcomes and is preserved as the best-learned individual in the society than through learners experiencing genomic processes like mutation, crossover, and selection. TLBO is successfully applied to many real-world problems [7, 8] due to its modest idea and high proficiency, and it emerged as a very pretty optimization technique. In general, TLBO yields better results to the other EC methods.

In this paper, an orthogonal design process is included in the elementary TLBO optimization. In this procedure, every student (learner) over the span of students in the orthogonal design is isolated into a few partial vectors where each of the students goes about as a factor. Orthogonal design, among all the different groupings, is

included to look through the finest scales. An orthogonal design process [9] demonstrates with orthogonal array (OA) and factor, however, an illustrative arrangement of courses of action for research to achieve respectable measures. The array is called orthogonal in light of the fact that all sections are assessed freely of each other, and the primary result of one factor does not inconvenience the evaluation of the fundamental impact of another factor. Experimental results proved that this technique is superior to the performance of the traditional EAs. To make orthogonal design faster and robust, TLBO is applied to an orthogonal design which is called OTLBO [10] and is implemented in this paper.

2 Preliminaries

Medical images are poorly illuminated, contain many uncertainties in the form of noise, having vague boundaries, gray levels are overlapped, invisible blood vessels, and difficult to extract objects from the image. Fuzzy sets in image processing increase contrast, flat the areas of interest, and refine the boundaries and fine erections of the image.

2.1 Fuzzy Sets

Fuzzy set I is an ordinary fuzzy set, characterized by a determinate regular $Y = \{y_1, y_2, y_3, \ldots, y_n\}$ factually as

$$I = \{(y, \mu_I(y))|y \in Y\} \tag{1}$$

The capacity $\mu_I(y) : Y \rightarrow [0, 1]$ speaks to the level of belongingness or the membership capacity of a component y in the limited set Y and $1 - \mu_I(y)$ represents the measure of non-belongingness or non-membership function.

Atanassov presented a novel fuzzy set the so-called intuitionistic fuzzy set that takes both membership function $\mu(y)$ and the non-membership function $v(y)$ into contemplation holding $\mu_I(y) \rightarrow [0, 1]$, $v_I(y) \rightarrow [0, 1]$. An intuitionistic fuzzy set I in a determinate set Y holding the situation $0 \leq \mu_I(y) + v_I(y) \leq 1$ is statistically denoted as

$$I = \{(y, \mu_I(y), v_I(y))|y \in Y\} \tag{2}$$

Atanassov also introduced a novel constraint $\pi_I(y)$ called the hesitation degree or intuitionistic fuzzy index, rising due to the absence of information or individual fault in conveying the membership degree. An IFS, I in a finite set Y with $\pi_I(y) + \mu_I(y) + v_I(y) = 1$, and the introduction of the hesitation degree may be represented as

$$I = \{(y, \mu_I(y), v_I(y), \pi_I(y))|y \in Y\} \tag{3}$$

where $0 \leq \pi_I(y) \leq 1$, for each $y \in Y$.

In this paper, an innovative objective function called IFE is introduced which plays a major part in image handling. Fuzzy entropy is calculated as in [11] and is utilized to improve the developed algorithm.

$$IFE(Z;\alpha) = \sum_{i=1}^{n} \pi_Z(y_i) \exp(1 - \pi_Z(y_i)) \tag{4}$$

where $\pi_Z(y_i) = 1 - (\mu_Z(y_i) + v_Z(y_i))$ denotes the hesitation degree, $\mu_Z(y_i)$ denotes the membership degree, and $v_Z(y_i)$ denotes the non-membership degree.

IFE utilizes Eq. (4) for designing and takes α standards going [0.1–1]. The extreme entropy standards are calculated by α optimization.

$$\alpha_{opt} = \max(IFE(Z;\alpha)) \tag{5}$$

The membership degrees of the IFIs are found by utilizing the known α values, and IFI is obtained as

$$F_{IFI} = \{(y,\ \mu_Z(y;\ \alpha),\ v_Z(y;\ \alpha),\ \pi_Z(y;\ \alpha))|y \in Y\} \tag{6}$$

2.2 TLBO

TLBO is a universal optimization technique which depends on population. It builds on the result of supervision of an educator based on the yield of students in a class. TLBO procedure imitates the teaching–learning capability of instructor and students in the teaching space. Based on its operation, this algorithm is classified into two portions: one is "teaching phase," where what a student learns from the teacher is explained. And the other is "learner phase," where what a student gains from interacting with other students is illustrated.

Learning from the teacher is termed as teacher phase. The parameter "mean" is considered here. The algorithm shifts the "mean" of learners toward its teacher in teacher phase by moving forward. Learning through collaboration with others is termed as learner phase, and mutual interface inclines to surge the understanding of the student. The arbitrary communication increases the learner's knowledge, and for one learner the additional learner is arbitrarily selected. The TLBO algorithm [6] is:

Step 1: Set the quantity of learners and iterations.
Step 2: Compute the mean of each variable and find the finest solution.
Step 3: Based on the best solution adjust the solution using

$$X_{new} = X_{old} + r_i(X_{old} - T_f\, M_e) \tag{7}$$

Step 4: Compare the solutions, if the new solution is not superior to present, then retains the former solution.

Step 5: Select two solutions randomly, on the off chance that the new arrangement is superior to existing. At that point, another arrangement is discovered utilizing.

$$Xnew^l_{(i)} = \begin{cases} X^l_m + rand\ *\ (X^l_m - X^l_n)\ if\ (f(X^l_{(i)}) < f(X^l_{(r)})) \\ X^l_m + rand\ *\ (X^l_n - X^l_m)\ otherwise \end{cases} \tag{8}$$

Step 6: The obtained new solution is superior to the previous solution and is taken as the best solution otherwise the previous solution is the best solution.

Step 7: After satisfying termination criteria, the final optimal coefficients are obtained. After completing the maximum number of iterations the algorithm is terminated.

2.3 Entropy

The degree of information content of an image represents entropy. Higher entropy signifies that an image contains maximum information content. It is defined as [12]:

$$H = -\sum_{i=0}^{L} h_{I_f}(i) \log_2 h_{I_f}(i) \tag{9}$$

where $h_{I_f}(i)$ speaks to the standardized histogram of the combined image and L means the measure of recurrence receptacles in the histogram.

3 The Proposed Image Fusion Based on OTLBO

Orthogonal design, developed as a logical intends to consider multifactor and multilevel issues. Orthogonal design abstracts to process orthogonal gathering L of M lines, where each line implies a strategy to be computed. An orthogonal design-based administrator is engaged with the standard TLBO to accomplish a more exact arrangement. OTLBO is proposed for improving the current population and to enrich the fitness function when the learner gets updated. The flowchart for the OTLBO is presented in Fig. 1. The following is the procedure used for fusing multimodal medical images, and the schematic of the developed medical image combination arrangement is shown in Fig. 2.

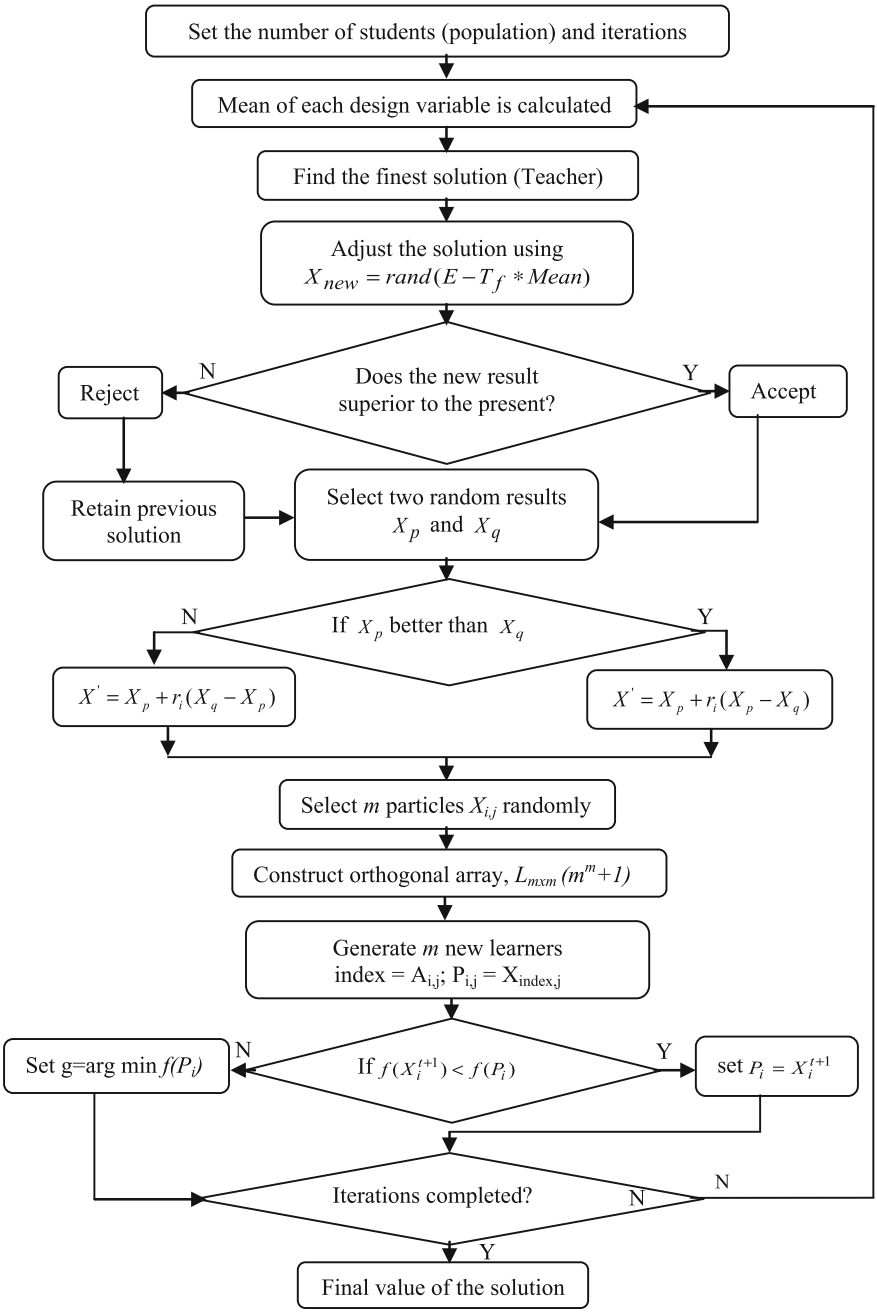


Fig. 1 Flowchart for orthogonal teaching–learning-based optimization (OTLBO)

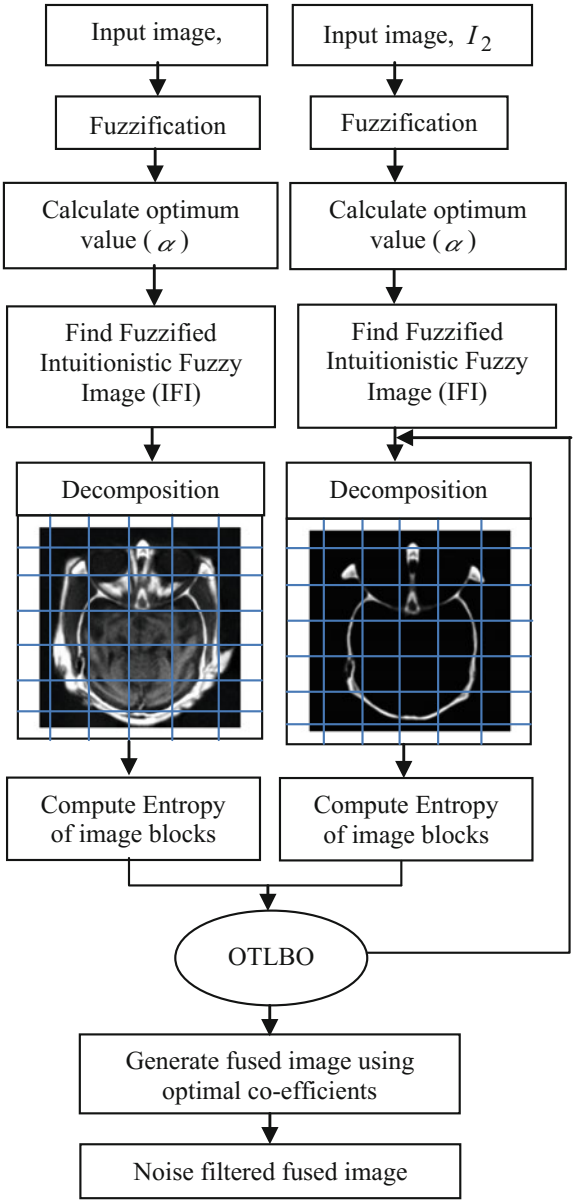


Fig. 2 Schematic of the proposed medical image combination scheme

(1) Read two input registered multimodal medical images.
(2) Utilizing below Eq. (10), the fuzzified first image I_1 of size $M \times N$ is

$$\mu_{Z1}(I_{ij1}) = \frac{I_{ij1} - l_{\min}}{l_{\max} - l_{\min}} \tag{10}$$

where I_{ij1} characterizes the dim level of the primary image and assortments from 0 to $L - 1$ (L is the outrageous dark-level esteem). $l_{\min}$ and $l_{\max}$ speak to the base and most extreme dark-level estimations of the primary image.
(3) Calculate the optimum value of α using entropy Eqs. (4) and (5) for the first image and this α varies for different input images.
(4) With the optimum value of α, find fuzzified IFI for the input image I_1 using beneath equations and indicate as I_{Y1}.

$$\mu_{IFI1}(I_{ij1}) = \mu_{Z1}(I_{ij1}) \tag{11}$$

where μ_{IFI1} is the membership degree of the first input image.

$$v_{IFI1}(I_{ij1};\, \alpha) = (1 - \mu_{IFI1}(I_{ij1};\, \alpha)^{\alpha})^{1/\alpha} \tag{12}$$

where $v_{IFI1}(I_{ij1};\, \alpha)$ is the optimized non-membership degree of the first image.

$$\pi_{IFI1}(I_{ij1};\, \alpha) = 1 - \mu_{IFI1}(I_{ij1};\, \alpha) - v_{IFI1}(I_{ij1};\, \alpha) \tag{13}$$

where $\pi_{IFI1}(I_{ij1};\, \alpha)$ is the optimized hesitation degree of the first image.

$$I_{Y1} = \left\{(I_{ij1},\ \mu_{IFI1}(I_{ij1};\, \alpha),\ v_{IFI1}(I_{ij1};\, \alpha),\ \pi_{IFI1}(I_{ij1};\, \alpha))\right\} \tag{14}$$

(5) Repeat the above procedure from step 2 to step 4 for the second input image, I_2 to find I_{Y2}.

$$I_{Y2} = \left\{(I_{ij2},\ \mu_{IFI2}(I_{ij2};\, \alpha),\ v_{IFI2}(I_{ij2};\, \alpha),\ \pi_{IFI2}(I_{ij2};\, \alpha))\right\} \tag{15}$$

where $\mu_{IFI2}(I_{ij2};\, \alpha)$, $v_{IFI2}(I_{ij2};\, \alpha)$, $\pi_{IFI2}(I_{ij2};\, \alpha))$ are optimized membership, non-membership and hesitation degrees of the second image, respectively.
(6) Images I_{Y1} and I_{Y2} are decayed into $m \times n$ windows, and in this algorithm, 5×5 is the window size taken and then the entropy of each block of two images is computed separately.
(7) Then, perform OTLBO algorithm to find optimal coefficients.
(8) Lastly, utilizing these optimal coefficients, the final fused image is attained, and the fused image is filtered for noise removal.

4 Implementation and Experimental Results

The execution of the proposed technique on various sets of multimodal medicinal images is presented and is compared with Jin and Wang [8] method. Here, all the pairs of multimodal medical images are registered and are available in [13–15]. The algorithm developed and implemented in MATLAB 8.1 [16].

The first example shown in Fig. 3-1a, b discourses CT and MRI images with a size of 256 × 256. The CT image provides information about bones and hard tissues while the MRI image gives soft tissue information. Fusing these two images provide abundant information in a single image, which helps to diagnose a disease. Figure 3-1d is the merged image of the proposed system, which visually proves that the image is of higher contrast and luminance than the fused image of the existing procedure [8]. Table 1 gives the independent evaluation of fusion results for MRI and CT images. Further comparisons of the results are performed using objective criteria [12] such as PSNR measured in dB (decibels), UQI, SSIM, CC, E is measured in bits per pixel, SF is measured in cycles per millimeter, $Q^{AB/F}$, and SD is measured in pixel intensity. PSNR (65.18) that is large for the proposed method than PSNR (61.23) of the proposed method. SF, SD, bits per pixel, UQI, SSIM, CC are large compared to the existing method. It is observed from the figure that the proposed method provides better performance regarding contrast, luminance, and visibility of the fused image.

The second case addresses a T1-weighted MR picture and the MRA picture with white structures as some ailment appeared in Fig. 3-2a, b. The combination of these two pictures gives correlative data inside a solitary picture, which is useful for better therapeutic diagnostics. Table 1 gives the target assessment of combination outcomes for T1-weighted MR and the MRA picture. The proposed method has large PSNR of 69.73, and correlation coefficient is 0.892. Figure 3-2d provides the fused image for the proposed method with high spatial resolution having a pixel intensity of 79.44.

The third illustration tends to MRI and PET pictures appeared in Fig. 3-3a, b. Table 1 gives the objective evaluation of fusion results for MRI and PET images. The fused image of the developed algorithm has PSNR of 69.93, UQI of 0.74, SSIM of 0.824, CC of 0.863, bits per pixel of 6.635, SF of 44.24, edge information of 0.767, and pixel density of 70.64. The fused image gives detailed information about the size of the disease present, which is not visible in other fused images and which is better diagnosed by doctors.

The fourth course of action of pictures is MRI and SPECT brain tumor pictures appeared in Fig. 3-4a, b. It is seen from the Fig. 3-4d that the combined picture of the proposed strategy contained by the tumor is evidently enhanced when contrasted with the existing technique. Table 1 gives the objective evaluation of fusion results for MRI and SPET images. The fused image has a pixel intensity of 52.64, and edge information transferred is 0.673 with large PSNR of 67.49. It is clearly observed that high accuracy is attained by the developed algorithm. Efficiently it abstracts the complementary and redundant information from MRI and SPECT images thereby producing an extremely consistent fused output image for detection of the tumor.

Table 1 Performance comparison of fused images of TLBO algorithm [8] and the proposed OTLBO algorithm

Modality pair	Fusion technique	PSNR	UQI	SSIM	CC	E	SF	$Q^{AB/F}$	SD
CT–MRI	TLBO [8]	61.23	0.762	0.912	0.913	6.294	21.442	0.711	46.85
	OTLBO	65.18	0.798	0.998	0.948	6.703	27.29	0.859	90.76
MR–MRA	TLBO [8]	64.33	0.667	0.783	0.814	5.926	25.73	0.631	49.45
	OTLBO	69.73	0.823	0.846	0.892	6.968	41.36	0.751	79.44
MRI–PET	TLBO [8]	62.16	0.776	0.801	0.835	6.273	43.18	0.625	50.37
	OTLBO	69.93	0.784	0.824	0.863	6.635	44.24	0.767	70.64
MRI–SPECT	TLBO [8]	65.78	0.48	0.841	0.863	5.595	35.63	0.408	21.57
	OTLBO	67.49	0.771	0.935	0.875	6.618	43.56	0.673	52.64
X-ray–VA	TLBO [8]	61.28	0.714	0.813	0.837	6.796	39.51	0.775	39.02
	OTLBO	67.69	0.882	0.915	0.872	7.185	53.64	0.872	48.82
PET–CT	TLBO [8]	65.75	0.781	0.889	0.874	6.268	35.49	0.734	43.08
	OTLBO	67.63	0.855	0.945	0.887	6.765	44.65	0.758	83.59

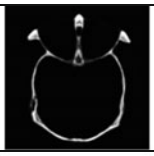
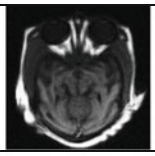
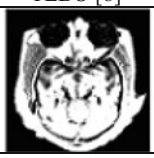
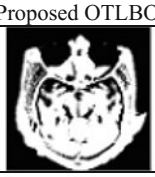
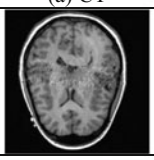
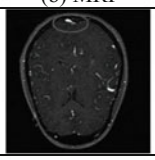
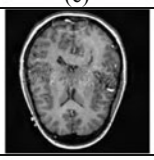
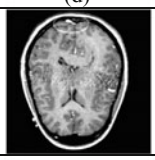
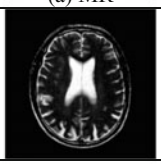
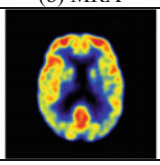
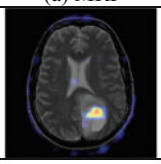
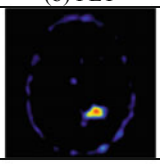
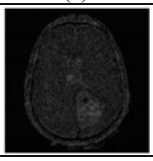
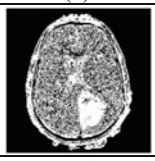
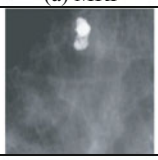
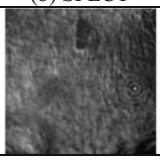
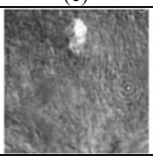
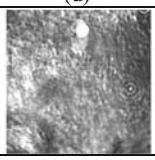
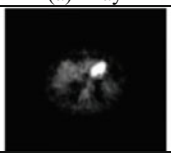
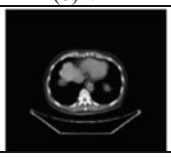
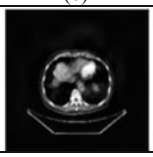
S.No	Modality 1	Modality 2	Fused image by TLBO [8]	Fused image by Proposed OTLBO
1.				
	(a) CT	(b) MRI	(c)	(d)
2.				
	(a) MR	(b) MRA	(c)	(d)
3.				
	(a) MRI	(b) PET	(c)	(d)
4.				
	(a) MRI	(b) SPECT	(c)	(d)
5.				
	(a) Xray	(b) VA	(c)	(d)
6.				
	(a) PET	(b) CT	(c)	(d)

Fig. 3 Fusion results of multimodal medical images using TLBO algorithm [8] and the proposed OTLBO algorithm

The fifth example tends to X-ray and vibro-acoustography VA pictures as shown in Fig. 3-5a, b. Table 1 gives the target assessment of combination outcomes for X-ray and VA pictures. Thus, by fusing two images, the diagnostic values of two modality images are more than individual images. The fused image has PSNR of 67.69 larger

than that of existing method. It has a high average number of bits per pixel of 7.185 for the proposed method which gives more data content than the current technique.

The last case speaks to multimodality images, for example, the PET and CT image appeared in Fig. 3-6a, b. The PET picture shows the cerebrum work, has a low spatial assurance. A CT picture gives thick structures like bones and embeds with less bending; it cannot recognize practical changes. Intertwining these two pictures, we get both practical information and extra spatial highlights without spatial mutilation and this is found in Fig. 3-6d for the developed algorithm.

5 Conclusions

This paper presents the optimal block size for the fusion of multimodal medical images by a global optimization procedure called OTLBO. The optimum solution is achieved by an efficient search of a large sample space using orthogonal design. Here, for a fixed number of iterations, algorithm is run and optimal coefficients are obtained. So, finally the final fused image is acquired. The obtained results are compared with existing TLBO with respect to different quantitative measures. From the result analysis, it is obvious that OTLBO outstrips the existing method in terms of accuracy. We can achieve that OTLBO is an intense technique in advancing the blocks that are discrete and multimodal in giving quality ideal outcomes.

References

1. Yang, Y., Park, D.S., Huang, S., Rao, N.: Medical image fusion via an effective wavelet based approach. EURASIP J. Adv. Signal Proc. (2010)
2. Gholam, H.H., Alizad, A., Fatemi, M.: Integration of Vibro-acoustography imaging modality with the traditional mammography. Int. J. Biomed. Imaging. Hindawi Publishing Corporation (2007)
3. Baum, K.G., Raerty, K., Helguera, M., Schmidt, E.: Investigation of PET/MRI image fusion schemes for enhanced breast cancer diagnosis. In: Proceedings of IEEE Seventh Symposium Conference on Nuclear Science (NSS) (2007)
4. Zadeh, L.A.: Fuzzy sets. Information. Control **8**, 338–353 (1965)
5. Atanassov, K.T.: Intuitionistic fuzzy sets. Fuzzy Sets Syst. **20**, 87–96 (1986)
6. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. **43**(3), 303–315 (2011)
7. Vedat, T.: Design of planar steel frames using teaching-learning based optimization. Eng. Struct. **34**, 225–232 (2012)
8. Jin, H., Wang, Y.: A fusion method for visible and infrared images based on contrast pyramid with teaching learning based optimization. Infrared Phys. Technol. **64**, 134–142 (2014)
9. Fang, K.T., Ma, C.X.: Orthogonal and Uniform Design. Science Press, Beijing (2001)
10. Suresh, C.S.: A teaching learning based optimization based on orthogonal design for solving global optimization problems. Springer Plus **2**, 130 (2013)
11. Chaira, T.: A novel intuitionistic fuzzy c means clustering algorithm and its application to medical images. Appl. Soft Comput. **11**(2), 1711–1717 (2011)

12. Jagalingam, P., Hegde, A.V.: A review of quality metrics for fused image. In: Aquatic Procedia of International Conference on Water Resources, Coastal and Ocean Engineering (ICWRCOE) (2015)
13. Tirupal, T., Mohan, B.C., Kumar, S.S.: Multimodal medical image fusion based on sugeno's intuitionistic fuzzy sets. ETRI J. **39**(2), 173–180 (2017)
14. www.med.harvard.edu/aanlib
15. www.metapix.de
16. www.mathworks.in

Inspection–Repair-Based Availability Optimization of Distribution System Using Bare Bones Particle Swarm Optimization

Aditya Tiwary

Abstract Maintenance is the crucial part for any engineering systems. The performance of engineering systems can be improved by proper maintenance at proper interval of time. Maintenance at proper interval of time is required for the distribution system which is an important subsystem of the power system, to enhance its efficiency and its proper utilization by the customers. In view of the above, in this paper bare bones PSO (BBPSO)-based technique is developed for optimizing inspection- and repair-based availability of distribution systems. The technique is proposed for evaluating the optimum time duration between two inspections of the distribution system with respect to cost function. The result obtained by BBPSO is compared with results of different variants of PSO such as basic particle swarm optimization (PSO), evolutionary particle swarm optimization (EPSO), and adaptive particle swarm optimization (APSO). Radial and meshed distribution systems were used for implementing the above algorithm.

Keywords Reliability · Availability · Bare bones particle swarm optimization Failure rate · Radial and meshed distribution systems

List of Symbols

T_{cycle}	cycle time
t_{in}	inspection duration
t_{ert}	repair time expected after the inspection
τ_d	period between inspections
c_i	cost coefficient for ith section
τ_i	interval between two inspection–repair for ith section
NC	number of sections

A. Tiwary (✉)
Electrical & Electronics Engineering Department,
Institute of Engineering & Science, IPS Academy, Indore, India
e-mail: raditya2002@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_38

NLP	load points total number
U_k	kth load point unavailability
U_{d-k}	unavailability threshold value at kth load point
A_K	kth load point availability
$A(\tau_d)$	availability for each section
t_{dn}	downtime
t_{repair}	repair time
$R(\tau_d)$	reliability function

1 Introduction

Electric power system is extremely complex in nature. This is because of its large physical size; geographically widely dispersed, international, and national interconnections; naturally followed physical laws; and many more reasons. Distribution system is a important subsystem of the complex electrical power system. Their main aim is on providing an economic and reliable electrical energy to the customers [1]. Various methods of reliability evaluation have been discussed in literature [1–10]. Different reliability calculations were proposed by Gaver et al. [11]. Patton [12] proposed reliability performance indices calculation technique for system. Probability distributions related to distribution system reliability were discussed in [13, 14]. Composite system reliability evaluation computational tool was proposed by Pereira and Pinto [15]. Reliability design of distribution system was developed by Su and Lii [16]. Meziane et al. [17] proposed ant colony algorithm-based reliability optimization. Allan and Silva [18] described a model and technique for evaluating the reliability indices in a meshed network. The methodology of using a database management system in distribution system reliability evaluation was developed by Hsu et al. [19]. Longo and Puntel [20] presented value-based reliability planning procedures for evaluation of distribution system enhancement. Sallam et al. [21] proposed optimal reliability indices for electrical distribution system. Heydt and Graf [22] discussed reliability evaluation using MCS. Ortmeyer et al. [23] presented reliability-based distribution system design. Wang and Li [24] proposed a method for reliability evaluation of distribution system. Billinton and Acharya [25] proposed weather-based distribution system reliability evaluation. Carpaneto and Chicco [26] proposed a method for evaluation of probability density function of distribution system reliability indices. Xie et al. [27] proposed reliability evaluation of electrical distribution system. Bie et al. [28] proposed reliability evaluation method based on MCS. Dehghanian et al. [29] proposed a scheme for reliability-centered maintenance in distribution power system. Parallel MCS for distribution system is discussed [30]. Huda and Zivanovic [31] developed accelerated distribution system reliability calculation. Reliability evaluation of distribution system with renewable DG is done by Adefarati and Bansal [32]. Optimal reconfiguration of electrical system was proposed by Lopez et al. [33]. Optimal placement of switches for reliability improvement was proposed by Ray et al. [34]. Tiwary et al. [35] developed a methodology for opti-

mizing availability of distribution systems using teaching–learning algorithm. Arya et al. [36] proposed a method for evaluating reliability indices considering random repair time omission using MCS. Tiwary et al. [37] developed a method for determination of optimum period between inspections for distribution system. Arya et al. [38] proposed smooth bootstrapping-based method for evaluating reliability indices. Tiwary et al. [39] developed a method for determining reliability indices using state transition sampling technique.

Vlachogiannis and Lee [40] proposed PSO-based method for power system. Kerdphol et al. [41] proposed optimum sizing of battery using PSO. Bhattacharya and Raj [42] have developed PSO-based bio-inspired algorithm for reactive power planning. Optimization of a battery energy storage system using PSO was proposed by Kerdphol et al. [43]. Banerjee et al. [44] discussed a method based on PSO technique for short-term hydro-wind-thermal scheduling. A PSO-based technique to find the near optimal solution for the capacitor allocation was developed by Ramadan et al. [45]. A comprehensive optimization technique for optimal placement and control of delta connected switched capacitor is proposed [46]. Hashemi et al. [47] proposed an algorithm concerning load reduction-based demand response. Huo et al. [48] proposed a methodology for optimal operation of interconnected energy hubs by using decomposed hybrid particle swarm and interior-point approach. A PSO algorithm for economic power dispatch was proposed by Kumar et al. [49]. Jadoun et al. [50] developed a modulated PSO for economic emission dispatch. Basu [51] developed a modified PSO for non-convex economic dispatch. Muniyasamy et al. [52] proposed an epidemic algorithm-based optimal power flow in electric grids. Kefi et al. [53] discussed the impact of ant size on ant supervised by PSO, AS-PSO performances. Sattianadan et al. [54] proposed power loss minimization by the placement of DG using PSO. Khatavkar et al. [55] obtained short-term estimation of transmission reliability margin using artificial neural network.

This paper has presented a methodology for obtaining optimum duration for each and every feeder section. The optimum duration is calculated between two maintenances with cost function minimization. The availability constraints have also been satisfied. The problem is solved by using BBPSO. Further, the results are compared with different variants of PSO such as basic particle swarm optimization (PSO), evolutionary particle swarm optimization (EPSO), and adaptive particle swarm optimization (APSO), and statistical inference has been provided.

2 Problem Formulation

Availability model based on inspection maintenance is developed in [35], for the feeder section of the distribution system.

$$T_{cycle} = \tau_d + t_{in} + t_{ert} \tag{1}$$

Downtime is characterized as follows

$$t_{dn} = t_{in} + t_{repair} \tag{2}$$

Expression for the expected repair time is described as [35]

$$t_{ert} = [1 - R(\tau_d)]t_{repair} \tag{3}$$

or

$$t_{ert} = [1 - R(\tau_d)] \cdot t_{dn} \tag{4}$$

The equation for the availability of the feeder section is

$$A(\tau_d) = \frac{\int_0^{\tau} R(t)dt}{\tau_d + t_{in} + t_{ert}} \tag{5}$$

or

$$A(\tau_d) = \frac{1 - \exp(-\lambda\tau_d)}{\lambda[\tau_d + t_{in} + t_{ert}]} \tag{6}$$

Assume failure rate as λ.

Objective function is described as follows [35]

$$J = \sum_{i=1}^{NC} c_i/\tau_i \tag{7}$$

The objective function in above equation describes the relation between the cost function and the duration between two inspection–repair of each and every feeder. The objective function will be useful in identifying the optimal duration between two repairs for feeder sections of the distribution system.

The above objective is minimized in order to obtain the optimum value of maintenance.

Constraints are considered as follows

$$U_k \leq U_{d-k} \qquad k = 1 \ldots NLP \tag{8}$$

In order to obtain load point unavailability, the following relation is used:

$$U_K = 1 - A_K \tag{9}$$

3 Different Techniques in Brief

3.1 Particle Swarm Optimization (PSO) Approach

PSO was proposed by Kennedy and Eberhart [56]. PSO is a population-based algorithm and is developed from social behavior of bird flocking. Each and every particle searches for the best solution in the search space. The particle approaches to better solution by the guidance provided by velocity and position in each iteration. The *i*th particle position is given as

$$S_i = \{s_{i1},\ s_{i2},\ \ldots,\ s_{in}\} \qquad i = 1,\ \ldots,\ m \tag{10}$$

where m represent population size.

The n-dimensional particle velocity is as follows

$$\rho_i = \{\rho_{i1},\ \rho_{i2},\ \ldots,\ \rho_{in}\} \qquad i = 1,\ \ldots,\ m \tag{11}$$

Position and velocity of particle is updated as follows

$$\rho_i^{iter} = w\rho_i^{iter-1} + c_1 rand_1(P_{best}^{iter-1} - S_i^{iter-1}) + c_2 rand_2(G_{best}^{iter-1} - S_i^{iter-1}) \tag{12}$$

$$S_i^{iter} = S_i^{iter-1} + \rho_i^{iter} \tag{13}$$

where w is termed as inertia weight.

The inertia weight w is varied as

$$w^{iter} = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} * iter \tag{14}$$

where $iter_{\max}$ is maximum iteration specified, $w_{\max}$ and $w_{\min}$ are maximum and minimum values of inertia weight.

3.2 Bare Bones Particle Swarm Optimization (BBPSO) in Brief

In 2003, BBPSO was proposed by Kennedy [57]. In BBPSO, each particle only has a position vector and the velocity vector is eliminated. The new position is updated by using the Gaussian distribution.

The updated position equation is as follows

$$x_i^{(t+1)} = N(\frac{G_{best} + P_{best}}{2}, |G_{best} - P_{best}|) \tag{15}$$

N—Gaussian distribution, mean $\mu = (\frac{G_{best} + P_{best}}{2})$, standard deviation $\sigma = |G_{best} - P_{best}|$ for each particle.

3.3 Evolutionary Particle Swarm Optimization (EPSO) in Brief

EPSO is applied in Bhattacharyya et al. [58] for reactive power planning. Positions and velocities are modified as following

$$S_i^{iter+1} = S_i^{iter} + \rho_i^{iter+1} \tag{16}$$

$$\rho_i^{iter+1} = w_i \cdot \rho_k + w_{i1}(P_{besti} - S_i^{iter}) + w_{i2}(g_{best} - S_i^{iter}) \tag{17}$$

where

$$w_{ik} = w_{ik} + \tau \cdot N(0,\ 1)$$

$$g_{best} = g_{best} + \tau' \cdot N(0,\ 1)$$

3.4 Adaptive PSO Algorithm (APSO) in Brief

APSO is used for application in literature [58]. Velocity of the improved PSO is expressed as

$$\rho_i^{iter+1} = w_i + C_1 \cdot rand_3 * (P_{besti} - S_i^{iter}) + C_2 \cdot rand_4 * (g_{best} - S_i^{iter}) \tag{18}$$

In APSO, the search trajectory is controlled by two parameters P1 and P2 whose values are kept to 0.5 or higher.

4 Availability Optimization Using BBPSO: Computational Algorithm

Implementation of algorithm:

1. Generate swarm of m particles:
   $S_i^0 = \{X_{i1}^0,\ X_{i2}^0,\ \ldots,\ X_{iNC}^0\}$
   Initial component X_{i1}^0:
   $x_{i-j}^0 = (x_{j-\max} - x_{j-\min})rand + x_{j-\min}$
2. Set iter = 1
3. Obtain A_k and U_k at each load point
4. Check for inequality constraints
5. Obtain $J(X_i^{(0)})$. Identify the best position (P_{best}). Identify best individual of whole swarm
6. Update position by using Eq. (15)
7. Apply inequality constraints $U_k \leq U_{d-k}$. Apply bounce back technique to bring component within limit [59]
8. Evaluate global optimum solution. Obtain value for current iteration.
   $iter = iter + 1$
9. If $iter > iter_{\max}$
   Stop, otherwise repeat from step 5.

5 Results and Discussions

The developed PSO, BBPSO, EPSO, and APSO are implemented on two systems.

5.1 Case A—Radial Distribution System

Table 1 and Table 2 provide initial values of the radial distribution system and control parameters used in optimization algorithms PSO, BBPSO, EPSO, and APSO, respectively.

Table 3 gives optimum duration and optimum availability. 3.71% improvement in optimum duration is obtained for distribution segment 1 by BBPSO as compared to PSO algorithm. For distribution segments 2, 3 and 4, 3.87, 3.61 and 3.52% improvement is obtained by using BBPSO in comparison with PSO algorithm. For distribution segment 7, APSO 3.04, EPSO 2.94, and BBPSO 2.83% improvement is obtained in comparison with PSO method. A better value of optimum duration in hours has been provided by the variants of PSO in comparison with the PSO technique. Table 4 provides optimum value of availability using BBPSO, EPSO,

Table 1 Initial system data for radial distribution system [35]

Distribution segment	#1	#2	#3	#4	#5	#6	#7
λ/year $\times 10^{-5}$	0.04	0.02	0.03	0.05	0.02	0.01	0.01
c_i (Rs.) $\times 10^7$	2.00	3.00	4.00	3.50	4.50	5.00	1.00
$\tau_{j,\min}$ (h)	500	500	500	500	500	500	500
$\tau_{j,\max}$ (h)	6000	6000	6000	6000	6000	6000	6000
t_{ins} (h)	12	10	15	20	10	18	18
t_{ri} (h)	36	20	25	30	20	24	24

Table 2 Control parameters for APSO, EPSO, BBPSO, and PSO for radial distribution system for the best run

S. No.	Parameters	Values of parameters			
		APSO	EPSO	BBPSO	PSO
1	Population size	20	20	20	20
2	c_1	1.2	NA	NA	1.2
3	c_2	0.12	NA	NA	0.12
4	Maximum generation specified ($k_{\max}$)	500	500	500	500

Table 3 Optimal interval between inspection–repair for radial distribution system segments

Distribution segments	APSO		EPSO		BBPSO		PSO	
	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$
#1	4708	0.9965	4705	0.9965	4658	0.9965	4485	0.9964
#2	4736	0.9974	4730	0.9974	4728	0.9974	4545	0.9974
#3	4692	0.9961	4685	0.9961	4670	0.9961	4501	0.9960
#4	5152	0.9949	5146	0.9948	5105	0.9948	4925	0.9947
#5	4695	0.9975	4692	0.9974	4689	0.9974	4418	0.9973
#6	4612	0.9959	4607	0.9959	4598	0.9959	4410	0.9957
#7	4629	0.9960	4624	0.9959	4619	0.9959	4488	0.9958

APSO, and PSO algorithm. Table 5 provides comparison between different statistics parameters obtained by different variants of PSO optimization technique. Average value of cost function has improved from 7.25% by APSO, 7.11% by EPSO, and 6.24% by BBPSO than in comparison with PSO. Coefficient of variation of cost function has improved by 15.28%, 15.05%, and 14.04% by APSO, EPSO, and BBPSO, respectively, in comparison with PSO algorithm.

Table 4 Optimal availability at load points for radial distribution system

Load points	Optimum availability (APSO)	Optimum availability (EPSO)	Optimum availability (BBPSO)	Optimum availability (PSO)
2	0.9965	0.9965	0.9965	0.9964
3	0.9940	0.9939	0.9939	0.9938
4	0.9901	0.9900	0.9900	0.9898
5	0.9915	0.9914	0.9913	0.9911
6	0.9889	0.9888	0.9887	0.9884
7	0.9899	0.9898	0.9898	0.9895
8	0.9860	0.9859	0.9858	0.9854

Table 5 Statistics obtained for minimum value of objective function with APSO, EPSO, BBPSO, and PSO based on 100 runs for radial distribution system

Parameters	Values of parameters (APSO)	Values of parameters (EPSO)	Values of parameters (BBPSO)	Values of parameters (PSO)
Average value of cost function (Rs.) $\times 10^4$	6.39	6.40	6.46	6.89
Standard Deviation of cost function (Rs.) $\times 10^3$	15.08	15.12	15.30	17.80
Coefficient of variation of cost function	0.2360	0.2364	0.2368	0.2583
Median value of cost function (Rs.) $\times 10^4$	5.95	5.97	6.01	6.12
Minimum value of cost function in 100 runs (Rs.) $\times 10^4$	4.36	4.38	4.40	4.41
Maximum value of cost function in 100 runs (Rs.) $\times 10^4$	9.82	9.83	9.85	9.99
Length of confidence interval (Rs.) $\times 10^3$	5.94	5.95	6.02	7.01
Upper bound on confidence interval (Rs.) $\times 10^4$	6.68	6.69	6.76	7.24
Lower bound on confidence interval (Rs.) $\times 10^4$	6.09	6.10	6.16	6.54
Frequency of convergence (f)	62	60	58	51

Table 6 CPU time required for convergence on Intel Core 2 Duo Processor, 2.10 GHz, for radial distribution system

Distribution system	CPU time (s) required using APSO	CPU time (s) required using EPSO	CPU time (s) required using BBPSO	CPU time (s) required using PSO
Radial	8.95	8.97	9.02	10.98

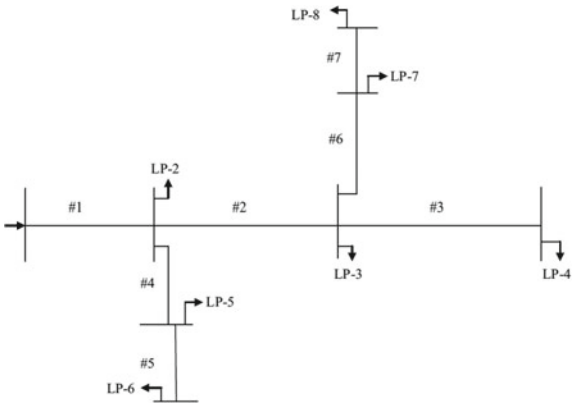


Fig. 1 Eight-node radial distribution system [36]

Evolution of best fitness value (objective function) by EPSO, APSO, BBPSO, and PSO is shown in Fig. 2. For the four algorithms, CPU time by APSO, EPSO, and BBPSO has reduced by 18.48%, 18.30%, and 17%, respectively, than in comparison with PSO algorithm as shown in Table 6 (Figs. 1 and 3).

5.2 Case B—Meshed Distribution System

Table 7 and Table 8 give initial values of the meshed distribution system and the control parameters used in optimization algorithms, respectively.

Table 9 shows the optimum duration and optimum availability. For distribution segments 1, 2, 3, 4, 5, 6, 7, 8 and 9, 7.40, 6.61, 5.25, 4.61, 2.33, 5.73, 4.57, 4.47 and 5.09% improvement is obtained by using BBPSO and 7.84, 7.14, 5.52, 4.73, 2.69, 5.92, 4.65, 4.78 and 5.52% improvement is obtained by using APSO, in comparison with PSO algorithm. For distribution segment 10, APSO 4.49, EPSO 4.30, and BBPSO 3.95% improvement is obtained in comparison with PSO method. As far as segments 16, 17 and 18 are concerned, APSO has reported 5.98, 6.55, and 4.59% improvement, EPSO has given 5.96, 6.34, and 4.53% improvement, and BBPSO has

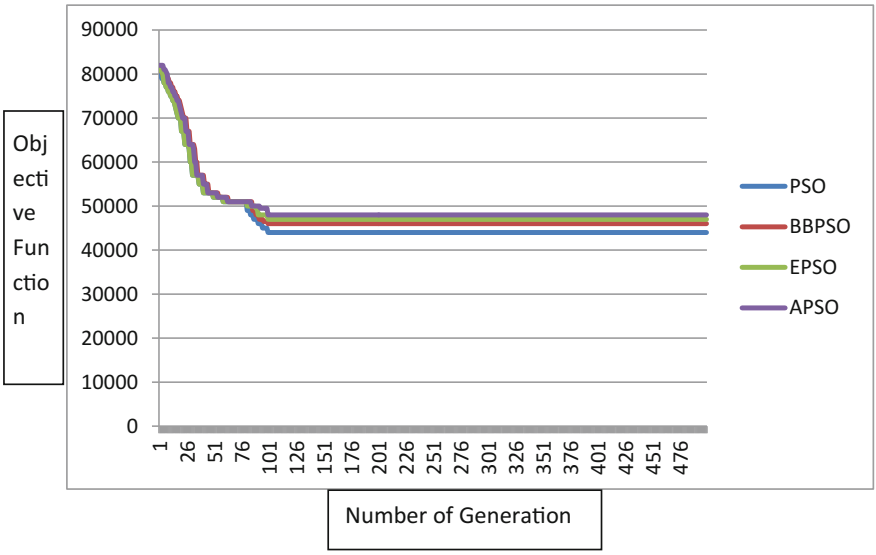


Fig. 2 Variation of best value of objective function with number of generations for radial distribution system

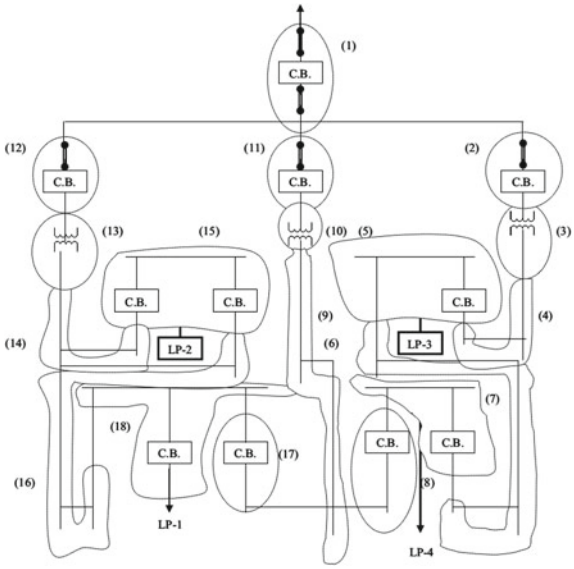


Fig. 3 Sample meshed distribution system [36]

reported an improvement of 5.78, 6.13, and 4.17% as compared to PSO algorithm. A better value of optimum duration in hours and optimum availability is been shown by the different variants of PSO in comparison with the basic PSO algorithm.

Table 7 Initial system data for meshed distribution system [35]

Distribution section	λ_i^0 failure/year $\times 10^{-5}$	c_i (Rs.) $\times 10^7$	$\tau_{j,\min}$ (h)	$\tau_{j,\max}$ (h)	t_{ins} (h)	t_{ri} (h)
#1	0.03	2.00	500	6000	10	30
#2	0.01	3.00	500	6000	15	20
#3	0.05	3.50	500	6000	12	23
#4	0.01	1.00	500	6000	20	28
#5	0.03	4.00	500	6000	18	40
#6	0.05	4.50	500	6000	16	38
#7	0.04	2.50	500	6000	13	36
#8	0.03	1.00	500	6000	10	33
#9	0.01	2.00	500	6000	17	26
#10	0.05	5.00	500	6000	18	28
#11	0.01	1.50	500	6000	20	31
#12	0.05	4.30	500	6000	11	21
#13	0.02	3.20	500	6000	16	38
#14	0.03	4.80	500	6000	18	39
#15	0.04	3.00	500	6000	20	34
#16	0.03	4.30	500	6000	10	32
#17	0.05	4.40	500	6000	15	38
#18	0.03	4.60	500	6000	10	29

Table 8 Control parameters for APSO, EPSO, BBPSO, and PSO for meshed distribution system

S. No.	Parameters	Values of parameters			
		APSO	EPSO	BBPSO	PSO
1	Population size	20	20	20	20
2	c_1	1.2	NA	NA	1.2
3	c_2	0.12	NA	NA	0.12
4	Maximum generation specified ($k_{\max}$)	500	500	500	500

Optimum value of availability is provided in Table 10. Table 11 provides different statistics parameters. Average value of cost function has improved from 15.20% by APSO, 14.03% by EPSO, and 11.11% by BBPSO than in comparison with PSO. Coefficient of variation of cost function has improved by 8.11%, 8.04%, and 7.78% by APSO, EPSO, and BBPSO, respectively, in comparison with PSO algorithm. Evolution of best fitness value is shown in Fig. 4. Table 12 provides CPU time by APSO, EPSO, and BBPSO has reduced by 13.32%, 12.98% and 12.07%, respectively, than in comparison with PSO algorithm.

Table 9 Optimal interval between inspection–repair for meshed distribution system segments

Distribution segments	APSO		EPSO		BBPSO		PSO	
	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$	Optimal duration, τ_i (h)	Optimum availability $A_i(\tau_i)$
#1	2510	0.9956	2508	0.9956	2498	0.9956	2313	0.9953
#2	2798	0.9946	2796	0.9945	2782	0.9945	2598	0.9941
#3	4889	0.9964	4881	0.9964	4875	0.9963	4619	0.9962
#4	5299	0.9960	5295	0.9960	5292	0.9960	5048	0.9958
#5	5239	0.9958	5237	0.9958	5220	0.9958	5098	0.9957
#6	5099	0.9956	5097	0.9956	5089	0.9956	4797	0.9955
#7	4599	0.9962	4598	0.9962	4595	0.9962	4385	0.9962
#8	4598	0.9971	4595	0.9971	4583	0.9971	4378	0.9971
#9	2898	0.9940	2894	0.9940	2885	0.9940	2738	0.9937
#10	4985	0.9951	4975	0.9951	4957	0.9951	4761	0.9950
#11	4999	0.9957	4996	0.9957	4971	0.9957	4792	0.9956
#12	4587	0.9964	4574	0.9964	4551	0.9964	4412	0.9964
#13	4658	0.9961	4656	0.9961	4648	0.9961	4502	0.9960
#14	4777	0.9955	4767	0.9955	4738	0.9955	4598	0.9954
#15	4548	0.9947	4539	0.9947	4524	0.9947	4375	0.9946
#16	4799	0.9972	4796	0.9972	4789	0.9972	4512	0.9971
#17	4519	0.9955	4509	0.9955	4499	0.9955	4223	0.9954
#18	4548	0.9971	4545	0.9971	4528	0.9971	4339	0.9970

Table 10 Optimal availability at load points for meshed distribution system

Load points	Optimum availability (APSO)	Optimum availability (EPSO)	Optimum availability (BBPSO)	Optimum Availability (PSO)
1	0.9927	0.9927	0.9926	0.9922
2	0.9905	0.9904	0.9902	0.9898
3	0.9918	0.9918	0.9913	0.9909
4	0.9922	0.9921	0.9917	0.9914

6 Conclusion

An innovative method is developed for optimum interval determination between two preventive maintenance activities. Availability constraints on load point have been accounted. Formulated optimization problem has been solved by using three variants of PSO, i.e., adaptive PSO (APSO), evolutionary PSO (EPSO), bare bones PSO (BBPSO), and the results have been compared to the results obtained by the basic PSO algorithm. The variants of PSO have shown a better result than in comparison with the PSO algorithm. The proposed methodology is implemented on radial and meshed distribution systems. Different statistical parameters such as average cost

Table 11 Statistics obtained for minimum value of objective function with APSO, EPSO, BBPSO, and PSO based on 100 runs for meshed distribution system

Parameters	Values of parameters (APSO)	Values of parameters (EPSO)	Values of parameters (BBPSO)	Values of parameters (PSO)
Average value of cost function (Rs.) $\times 10^4$	1.45	1.47	1.52	1.71
Standard deviation of cost function (Rs.) $\times 10^3$	1.98	2.01	2.09	2.55
Coefficient of variation of cost function	0.1370	0.1371	0.1375	0.1491
Median value of cost function (Rs.) $\times 10^4$	1.43	1.45	1.50	1.55
Minimum value of cost function in 100 runs (Rs.) $\times 10^4$	1.17	1.18	1.22	1.30
Maximum value of cost function in 100 runs (Rs.) $\times 10^4$	1.86	1.88	1.90	2.10
Length of confidence interval (Rs.) $\times 10^2$	7.80	7.91	8.23	10.04
Upper bound on confidence interval (Rs.) $\times 10^4$	1.48	1.50	1.56	1.76
Lower bound on confidence interval (Rs.) $\times 10^4$	1.43	1.45	1.47	1.65
Frequency of convergence (f)	63	61	60	49

Table 12 CPU time required for convergence on Intel Core 2 Duo Processor, 2.10 GHz, for meshed distribution system

Distribution system	CPU time (s) required using APSO	CPU time (s) required using EPSO	CPU time (s) required using BBPSO	CPU time (s) required using PSO
Meshed	10.41	10.45	10.56	12.01

function value, standard deviation of cost function, coefficient of variation of cost function have also been obtained by using the four optimization algorithm for the two test systems considered.

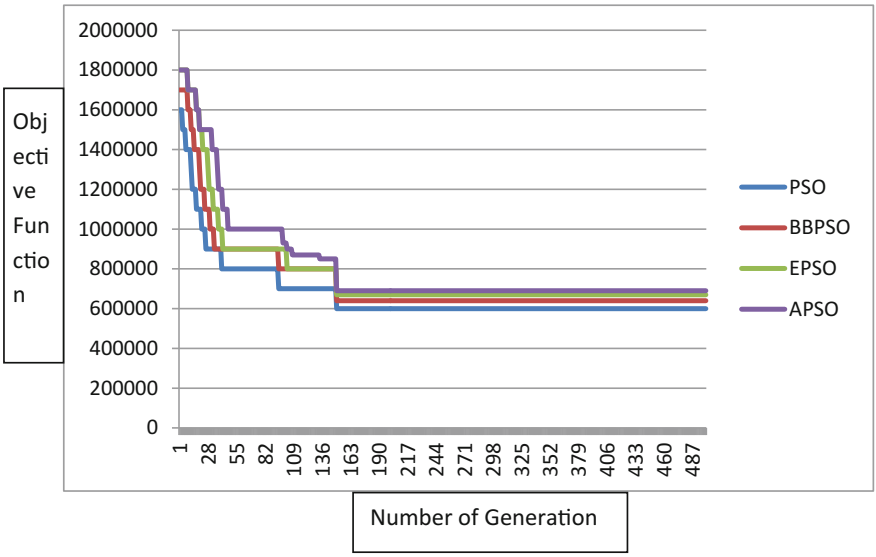


Fig. 4 Variation of best value of objective function with number of generations for meshed distribution system

References

1. Billinton, R., Allan, R.N.: Reliability Evaluation of Power System. Springer International Edition (1996)
2. Elmakias, D.: New Computational Methods in Power System Reliability. Springer International Edition (2000)
3. Gertsbakh, I.: Reliability Theory with Applications to Preventive Maintenance. Springer International Edition (2000)
4. Gangel, M.W., Ringlee, R.J.: Distribution system reliability performance. IEEE Trans. **PAS-1968 87**(7), 1657–1665 (1968)
5. IEEE Committee Report: Bibliography on the application of probability methods in power system reliability evaluation. IEEE Trans. Power Appar. Syst. **PAS-91**, 649–660 (1972)
6. IEEE Committee Report: Bibliography on the application of probability methods in power system reliability evaluation 1971–1977. IEEE Trans. Power Appar. Syst. **PAS-97**, 2235–2242 (1978)
7. Allan, R.N., Dialynas, E.N., Homer, I.R.: Modelling and evaluating the reliability of distribution systems. IEEE Trans. **PAS-98**, 2181–2189 (1979)
8. Billinton, R., Allan, R.N.: Reliability Evaluation of Engineering Systems. Springer International Edition (1992)
9. Pereira, M.V.F., Pinto, L.M.V.G.: A new computational tool for composite reliability evaluation. IEEE Trans. **PAS 7**(1), 258–263 (1992)
10. Allan, R.N., Billinton, R.: Probabilistic assessment of power systems. Proc. IEEE **88**(2), 140–162 (2000)
11. Gaver, D.P., Montmeat, F.E., Patton, A.D.: Power system reliability: I-measures of reliability and methods of calculation. IEEE Trans. Power Appar. Syst. **PAS-83**, 727–737 (1964)
12. Patton, A.D.: Probability distribution of transmission and distribution reliability performance indices. In: Reliability Conference for the Electric Power Industry, pp. 120–123 (1973)

13. Billinton, R., Wojczyuski, E., Rodych, V.: Probability distributions associated with distribution system reliability indices. In: Reliability Conference for the Electric Power Industry (1980)
14. Billinton, R., Wojczyuski, E., Godfrey, M.: Practical calculations of distribution system reliability indices and their probability distributions. Trans. Can. Electr. Assoc. Paper 81-D-41 (1981)
15. Pereira, M.V.F., Pinto, L.M.V.G.: A new computational tool for composite reliability evaluation. IEEE Trans. Power Syst. **7**(1), 258–264 (1992)
16. Su, C.T., Lii, G.R.: Reliability design of distribution systems using modified genetic algorithms. Electr. Power Syst. Res. **60**, 201–206 (2002)
17. Meziane, R., Massim, Y., Zeblah, A., Ghoraf, A., Rahli, R.: Reliability optimization using ant colony algorithm under performance and cost constraints. Electr. Power Syst. Res. **76**, 1–8 (2005)
18. Allan, R.N., Da Silva, M.G.: Evaluation of reliability indices and outage costs in distribution systems. IEEE Trans. Power Syst. **10**(1), 413–419 (1995)
19. Hsu, Y.Y., Chen, L.M., Chen, J.L., Hsueh, M.C., Lin, C.T., Chen, Y.W., Chen, J.J., Liu, T.S.S., Chen, W.C., Chang, I.S.: Application of a microcomputer-based database management system to distribution system reliability evaluation. IEEE Trans. Power Delivery **5**, 343–350 (1990)
20. Longo, V., Puntel, W.R.: Evaluation of distribution system enhancements using value-based reliability planning procedures. IEEE Trans. Power Syst. **15**, 1148–1153 (2000)
21. Sallam, A.A., Desouky, M., Desouky, H.: Evaluation of optimal-reliability indices for electrical distribution systems. IEEE Trans. Reliab. **39**, 259–264 (1990)
22. Heydt, G.T., Graf, T.J.: Distribution system reliability evaluation using enhanced samples in a Monte Carlo approach. IEEE Trans. Power Systems. **25**, 2006–2008 (2010)
23. Ortmeyer, T.H., Reeves, J.A., Hou, D., McGrath, P.: Evaluation of sustained and momentary interruption impacts in reliability-based distribution system design. IEEE Trans. Power Delivery **25**, 3133–3138 (2010)
24. Wang, P., Li, W.: Reliability evaluation of distribution systems considering optimal restoration sequence and variable restoration times. IET Gener. Transm. Distrib. **1**, 688–695 (2007)
25. Billinton, R., Acharya, J.R.: Weather-based distribution system reliability evaluation. IEEE Proc. Gener. Trans. Distrib. **153**, 499–506 (2006)
26. Carpaneto, E., Chicco, G.: Evaluation of the probability density functions of distribution system reliability indices with a characteristic functions-based approach. IEEE Trans. Power Syst. **19**, 724–734 (2004)
27. Xie, K., Cao, K., Yu, D.C.: Reliability evaluation of electrical distribution networks containing multiple overhead feeders on a same tower. IEEE Trans. Power Syst. **26**, 2518–2525 (2011)
28. Bie, Z., Zhang, P., Li, G., Hua, B., Meehan, M., Wang, X.: Reliability evaluation of active distribution systems including microgrids. IEEE Trans. Power Syst. **27**, 2342–2350 (2012)
29. Dehghanian, P., Firuzabad, M.F., Aminifar, F., Billinton, R.: A comprehensive scheme for reliability centered maintenance in power distribution systems—Part I: methodology. IEEE Trans. Power Delivery **28**, 761–770 (2013)
30. Velasco, J.A.M., Guerra, G.: Parallel Monte Carlo approach for distribution reliability assessment. IET Gener. Transm. Distrib. **8**, 1810–1819 (2014)
31. Huda, A.S.N., Zivanovic, R.: Accelerated distribution systems reliability evaluation by multilevel Monte Carlo simulation: implementation of two discretisation schemes. IET Gener. Transm. Distrib. **11**, 3397–3405 (2017)
32. Adefarati, T., Bansal, R.C.: Reliability assessment of distribution system with the integration of renewable distributed generation. Appl. Energy **185**, 158–171 (2017)
33. Lopez, J.C., Lavorato, M., Rider, M.J.: Optimal reconfiguration of electrical distribution systems considering reliability indices improvement. Int. J. Electr. Power Energy Syst. **78**, 837–845 (2016)
34. Ray, S., Bhattacharya, A., Bhattacharjee, S.: Optimal placement of switches in a radial distribution network for reliability improvement. Int. J. Electr. Power Energy Syst. **76**, 53–68 (2016)

35. Tiwary, A., Arya, L.D., Arya, R., Choube, S.C.: Inspection repair based availability optimization of distribution systems using teaching learning based optimization. J. Inst. Eng. (India): Ser. B. **97**(3), 355–365 (2016)
36. Arya, L.D., Choube, S.C., Arya, R., Tiwary, A.: Evaluation of Reliability indices accounting omission of random repair time for distribution systems using Monte Carlo simulation. Int. J. Electr. Energy. System. **42**, 533–541 (2012)
37. Tiwary, A., Arya, R., Choube, S.C., Arya, L.D.: Determination of optimum period between inspections for distribution system based on availability accounting uncertainties in inspection time and repair time. J. Inst. Eng. (India): Ser. B (Springer) **93**(2), 67–72 (2012)
38. Arya, R., Tiwary, A., Choube, S.C., Arya, L.D.: A smooth bootstrapping based technique for evaluating distribution system reliability indices neglecting random interruption duration. Int. J. of Electr. Power Energy Syst. **51**, 307–310 (2013)
39. Tiwary, A., Arya, R., Choube, S.C., Arya, L.D.: Determination of reliability indices for distribution system using a state transition sampling technique accounting random down time omission. J. Inst. Eng. (India): Ser. B (Springer) **94**(1), pp. 71–83 (2013)
40. Vlachogiannis, J.G., Lee, K.Y.: A comparative study on particle swarm optimization for optimal steady State performance of power systems. IEEE Trans. Power Syst. **21**(4), 1318–1728 (2006)
41. Kerdphol, T., Qudaih, Y., Mitani, Y.: Optimum battery energy storage system using particle swarm optimization considering dynamic demand response for microgrids. Int. J. Electr. Power Energy Syst. **83**, 58–66 (2016)
42. Bhattacharyya, B., Raj, S.: PSO based bio inspired algorithms for reactive power planning. Int. J. Electr. Power Energy Syst. **74**, 396–402 (2016)
43. Kerdphol, T., Fuji, K., Mitani, Y., Watanabe, M., Qudaih, Y.: Optimization of a battery energy storage system using PSO for stand-alone microgrids. Int. J. Electr. Power Energy Syst. **81**, 32–39 (2016)
44. Banerjee, S., Dasgupta, K., Chanda, C.K.: Short term hydro-wind-thermal scheduling based on particle swarm optimization technique. Int. J. Electr. Power Energy Syst. **81**, 275–288 (2016)
45. Ramadan, H.S., Bendary, A.F., Nagy, S.: Particle swarm optimization algorithm for capacitor allocation problem in distribution system with wind turbine generators. Int. J. Electr. Power Energy Syst. **84**, 143–152 (2017)
46. Su, X., Masoum, A.S., Wolfs, P.J.: PSO and improved BSFS based sequential comprehensive placement and real-time multi-objective control of delta-connected switched capacitors in unbalanced radial MV distribution networks. IEEE Trans. Power Syst. **31**, 612–622 (2016)
47. Hashemi, S., Aghamohammadi, M.R., Sangrody, H.: Restoring desired voltage security margin based on demand response using load to source impedance ratio index and PSO. Int. J. Electr. Power Energy Syst. **96**, 143–151 (2018)
48. Huo, D., Le Blond, S., Gu, C., Wei, W., Yu, D.: Optimal operation of interconnected energy hubs by using decomposed hybrid particle swarm and interior-point approach. Int. J. Electr. Power Energy Syst. **95**, 36–46 (2018)
49. Kumar, R., Sharma, D., Sadu, A.: A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. Int. J. Electr. Power Energy Syst. **33**, 115–123 (2011)
50. Jadoun, V.K., Gupta, N., Niazi, K.R., Swarnkar, A.: Modulated particle swarm optimization for economic emission dispatch. Int. J. Electr. Power Energy Syst. **73**, 80–88 (2015)
51. Basu, M.: Modified particle swarm optimization for nonconvex economic dispatch problems. Int. J. Electr. Power Energy Syst. **69**, 304–312 (2015)
52. Muniyasamy, K., Srinivasan, S., Parthasarathy, S., Subathra, B., Dzitac, S.: Epidemic algorithm based optimal power flow in electric grids. Adv. Intell. Syst. Comput. **633**, 60–69 (2016)
53. Kefi, S., Rokbani, N., Alimi, A.M.: Impact of ant size on ant supervised by PSO, AS-PSO, performances. Adv. Intell. Syst. Comput. **552**, 567–577 (2017)
54. Sattianadan, D., Sudhakaran, M., Dash, S.S., Kumar, K.V., Biswal, B.: Power loss minimization by the placement of DG in distribution system using PSO. Adv. Intell. Syst. Comput. **199**, 497–504 (2013)
55. Khatavkar, V., Swathi, D., Mayodeo, H., Dharme, A.: Short term estimation of transmission reliability margin using artificial neural network. Adv. Intell. Syst. Comput. **628**, 17–27 (2017)

56. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948, Piscataway, NJ (1995)
57. Kennedy, J.: Bare bones particle swarms. In: Proceedings of IEEE Swarm Intelligence Symposium, pp. 80–87 (2003)
58. Bhattacharyya, B., Raj, S.: PSO based bio inspired algorithms for reactive power planning. Int. J. Electr. Power Energy Syst. **74**, 392–402 (2016)
59. Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Springer (2005)

Butterfly Optimizer for Placement and Sizing of Distributed Generation for Feeder Phase Balancing

Sujeet Mishra, Abhishek Kumar, Devender Singh and Rakesh Kumar Misra

Abstract The present work investigates a new strategy of applying single-phase distributed generations (DGs) for the problem of phase load balancing. The work demonstrates that single-phase DG can be applied at specific location and phase so that overall phase loads are balanced at the root node. The problem of suitable size, bus-phase location of DGs is a mixed integer nonlinear programming (MINLP) problem. This problem is attempted using butterfly optimizer to get solution.

Keywords Butterfly optimizer · Distributed generation · Feeder phase balancing Mixed integer non-linear programming

1 Introduction

Effective utilization of main substation transformer depends on the distribution of loads among the three phases. If the loads are unbalanced among the phases, then the phase having the maximum loading decides the MVA margin available on the main substation transformer. Thus, to improve the loading capability of the main substation, balancing of three phases is important. The load balancing is also seen from the perspective of sequence components. A balanced load will have balanced

S. Mishra
Research Designs and Standards Organization, Ministry of Railways, Lucknow, Uttar Pradesh, India
e-mail: mishrasujeet@gmail.com

A. Kumar (✉) · D. Singh · R. Kumar Misra
Department of Electrical Engineering, IIT (BHU) Varanasi, Varanasi 221005, Uttar Pradesh, India
e-mail: abhishek.kumar.eee13@iitbhu.ac.in

D. Singh
e-mail: dsingh.eee@iitbhu.ac.in

R. Kumar Misra
e-mail: rkmishra.eee@iitbhu.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_39

currents, and therefore, only positive-sequence currents are present in the system. The absence of negative- and zero-sequence currents ensures minimum flow in the neutral conductor, thus minimizing the neutral losses. The balancing of three phases can be obtained by using power electronic components [1]. The use of power electronic components increases the cost of the system. The load balancing could also be achieved by co-ordinating the voltage control equipment in the system to regulate the phase-wise current flows. Author of [2] used an optimal local flow concept to obtain the settings of voltage control equipment. The other method is to distribute the loads among the three phases through load-switching arrangements [3]. The load-switching arrangement assumes that at every possible load bus, there are arrangements to transfer load from one phase to the other. This poses an optimization problem of combinatorial nature having large number of combinations. In recent times, there has been a major focus on distributed generation. There are several types of DGs which can be interfaced to the grid for supplying renewable energy. In the present work, a proposal to use distributed generation to improve the system load balancing has been modelled. The present work aims towards utilizing single-phase DGs at selected buses to generate power in a given phase so that the overall load in the system is balanced. As mentioned earlier, the system of moderate size can explode in large number of combinations. The problem of load balancing is a type of problem which calls for search methods such as PSO, GA and other evolutionary algorithms. In the present work, butterfly optimizer [4] based search method has been used.

2 Problem Formulation

The problem formulation is discussed in three parts. In first part, mathematical representation of DGs switch is explained. DGs switch is basically an arrangement which can provide power from renewable sources to one of the phases of the bus. In second part, inclusion of DGs and its load model in the current injection-based three-phase distribution load flow is shown. Finally in third part, implementation of butterfly optimizer (BO) for optimal placement and sizing of DGs is discussed.

2.1 Mathematical Representation of DG Switch

The DG switch changeover for a particular DG, n, at kth iteration is described by the following vector.

$$S^k = \begin{pmatrix} B_n^k \\ Ph_n^k \\ Gp_n^k \\ Gq_n^k \end{pmatrix} \tag{1}$$

where $B_n^k \in (2, N)$, $Ph_n^k \in (1, 3)$, and $Gp_n^k + jGq_n^k$ is the complex power provided by the nth DG.

2.2 Inclusion of DGs in Three-Phase Distribution Load Flow

Modelling of DGs is generally done by negative load model in different applications and scenarios.

In this work, the current injection mismatch-based Newton–Raphson power flow algorithm is used and a similar approach followed as explained in [5]. To include the DGs in power flow algorithm, some modifications are made in [5].

In the current injection mismatch-based load flow algorithm, the load required is changed for the bus ($i == B_n^k$) at the phases ($j == Ph_n^k$) due to a nth DGs at k iteration and load required can be written in the following way,

$$p_i^{j,sp,k} = p_{i0}^{j,sp,k} - Gp_n^k, \quad q_i^{j,sp,k} = q_{i0}^{j,sp,k} - Gq_n^k \tag{2}$$

$j \in \{a, b, c\}$. A flat start for $|V_i^{j,k}|$s is taken. Using the voltages $|V_i^{j,k}|$, the loads can be calculated as

$$p_i^{j,cal,k} - jq_i^{j,cal,k} = \bar{V}_i^{*j,k}.\sum_{j=1}^{N}\sum_{m\in a,b,c} Y_{i,j}^m.\bar{V}_j^{m,k} \tag{3}$$

When the value of tolerance, ϵ, is smaller than a specified value, the load flow algorithm is considered as converged at a power flow solution, where ϵ is calculated by,

$$\epsilon = \max_{i\in all\ busses}\{|(p_i^{j,sp,k} - p_i^{j,cal,k}) - j(q_i^{j,sp,k} - q_i^{j,cal,k})|\} \tag{4}$$

The specified equivalent loads ($P_i^{j,sp,k}$ and $Q_i^{j,sp,k}$) at bus i, phase j, at the kth iteration can be obtained as the sum of powers being supplied to all the buses downstream.

2.3 Butterfly Optimizer-Based Phase Load Balancing

The original butterfly optimizer (BO) for unconstrained optimization problems was proposed in [4]. BO is a dual population-based nature-inspired optimization technique. In this section, a brief introduction of different processes of BO is discussed.

2.3.1 Dual Population of BO

In BO, there is dual population based on the position of male butterflies. First population, P_1, contains the perching position of every male butterfly. Population P_1 is defined as an N, D-dimensional vector. If k denotes the iteration, the P_1 at kth iteration consists of:

$$P_1^k = [\bar{x}_1^k, \bar{x}_2^k, \ldots, \bar{x}_N^k] \tag{5}$$

$$\bar{x}_i^k = [x_{i1}^k, x_{i2}^k, x_{i3}^k, \ldots, x_{iD}^k]^T, i = 1, 2, \ldots, N \tag{6}$$

where N is the number of male butterflies and D is the dimension of the problem.

Second population, P_2, contains the best perching position of every male butterfly. Population P_2 is defined as an N, D-dimensional vector. If k denotes the iteration, the P_2 at kth iteration consists of:

$$P_2^k = [m\bar{x}_1^k, m\bar{x}_2^k, \ldots, m\bar{x}_N^k] \tag{7}$$

$$m\bar{x}_i^k = [mx_{i1}^k, mx_{i2}^k, mx_{i3}^k, \ldots, mx_{iD}^k]^T, i = 1, 2, \ldots, N \tag{8}$$

The set of solution $m\bar{x}_i^k$ and $\bar{x}_i^k$ is associated with ith individual at kth iteration. BO uses the exploration and exploitation property of perching and patrolling mode, respectively, to search new solution for each $\bar{x}_i^k$ and $m\bar{x}_i^k$.

2.3.2 Initialization of Solutions

Initial dual solutions of each individual are randomly generated within the search space. Initialization of solutions is given by:

$$x_{ij}^0 = (U_j - L_j) * rand + L_j \tag{9}$$

$$mx_{ij}^0 = (U_j - L_j) * rand + L_j$$

where L_j and U_j are the lower and upper bound of jth parameter of the problem, respectively, and $rand$ is uniformly distributed random number between [0, 1].

2.3.3 Perching

Perching can be explained by dividing its function into two processes: *Crisscross modification* and *I-Selection*.

Crisscross modification
Crisscross modification produces a perching position vector, $\bar{x}_i^{k+1}$, by modifying one of its randomly selected element called as element d as follows:

$$x_{ij}^{k+1} = \begin{cases} \mathfrak{R}(x_{cc_i j}^k, mx_{cc_i j}^k) + F * (\mathfrak{R}(x_{g_i j}^k, mx_{g_i j}^k) - \mathfrak{R}(x_{r_i j}^k, mx_{r_i j}^k)), & \text{if } j == d \\ mx_{ij}^k, & \text{otherwise.} \end{cases} \tag{10}$$

where $\mathfrak{R}(*, *)$ is a random selection operator which selects one of the variables with equal probability, and cc_i is called as crisscross neighbour of ith butterfly. Index q_i

and r_i are randomly selected neighbours for ith butterfly. Random selection of index q_i and r_i is done in such a manner that they satisfy Eq. 11.

$$i \neq cc_i \neq q_i \neq r_i \tag{11}$$

At every iteration, before perching or patrolling, a crisscross vector, cc_i, of length N is calculated by re-shuffling the integers from 1 to N. The vectors $\bar{cc}$ consist of:

$$\bar{cc} = [cc_1, cc_2, cc_3, \ldots, cc_i]^T, i = 1, 2, 3, \ldots, N \tag{12}$$

I-Selection

I-Selection process generates the P_2, of next iteration. Selection of $\bar{x}_i^k$ in place of $\bar{mx}_i^k$ at $(k+1)$th iteration is decided by the objective function value. The individual having lower objective function value is selected as $\bar{mx_i}^k$ for the next iteration. Procedure of I-Selection can be given by Eq. 13

$$\bar{mx}_i^{k+1} = \begin{cases} \bar{x_i}^{k+1}, & \text{if } f(\bar{x_i}^{k+1}) \leq f(\bar{mx_i}^k) \\ \bar{mx_i}^k, & \text{otherwise.} \end{cases} \tag{13}$$

where $f(*)$ is the objective function value at $*$ position.

2.3.4 Patrolling

Male butterflies whose position is not updated during perching pass to patrolling mode. The process of patrolling is divided into two parts: *Towards-best modification* and *II-Selection*.

Towards-best modification
This process generates patrolling position vector, $\bar{ux}_i^{k+1}$

$$\bar{ux}_i^{k+1} = \bar{mx}_i^k + s * \bar{mv}_i^k + F * (\bar{mx}_{maxuv_i}^k - \bar{mx}_i^k) \tag{14}$$

where $\bar{mx}_{maxuv_i}$ is the most attractive position among population, F is a random value between [0, 1], and s is a constant between [0, 1].

II-Selection
II-Selection is same process as I-Selection, but only difference is that II-Selection also updates the velocity vector, $\bar{mv}_i^k$. Velocity vector updating is done in accordance with Eq. 15.

$$\bar{mv}_i^{k+1} = \begin{cases} \bar{ux}_i^{k+1} - \bar{mx}_i^k, & \text{if } f(\bar{ux_i}^{k+1}) \leq f(\bar{mx_i}^k) \\ d * \bar{mv}_i^k + F * (\bar{mx}_{maxuv_i}^k - \bar{mx}_i^k), & \text{otherwise.} \end{cases} \tag{15}$$

In every cycle/iteration, the solution of the optimization problem is the objective function value of position of most attractive male butterfly. Termination of BO algorithm depends upon two criteria: (i) The max number of function evaluation and (ii) convergence of solution; i.e. solution is not updated for specified number of consecutive iterations.

In the present work, BO has been applied to get the optimal location and sizing of DGs. The objective is to place the single-phase DGs in the system which provides real and reactive power at some of the buses among one of three phases resulting in minimum negative- and zero-sequence currents. A current unbalance index considering the zero-sequence components as well as the negative-sequence components has been used. The optimization problem for placement and sizing the single-phase DGs for minimum unbalance index at the main transformer, Root Mean Squared Current ($RMSI$), can be stated as follows.

$minimize$

$$RMSI(S) = (|I_0(1)|^2 + |I_2(1)|^2) \tag{16}$$

such that the load flow constraints are satisfied.

3 Test System and Studies Performed

The single line diagram, showing line type and line length (ft.), of a 25-bus, three-phase, test system [6] shown in Fig. 1 has been used for studying the load balancing proposed in this paper. The base load data and impedance matrices of this system for each bus are used as same as used in [6].

The parameters of BO algorithm used are as follows: $N = 100$, $F = 0.5$. An individual consists of 20 elements, and the first ten elements are integer codes taking values from 2 to 25 for first five elements and from 1 to 3 for next five elements.

The stopping criterion used is quite a stringent one, i.e. the BO is assumed to converge when for last 1000 iterations, (i) there is no change in $RMSI$, and (ii) $\|\bar{x}_k - \bar{x}_{k-1000}\|_1 \leq 0.0001$.

4 Results and Discussions

The result of optimization problem solved by BO is depicted in Table 1. The resulting phase-wise distribution of loads after including the DGs at their optimal position with optimal size calculated by BO is depicted in Table 2. It is observed that on the main substation, the load among the three phases gets balanced after re-phasing. The load flow results for base case system and system with DGs are given in Table 3. From Table 3, it is observed that the minimum system voltage in each of the phases ($Min(V^a)$, $Min(V^b)$, $Min(V^c)$) is low in base case for phase-a and phase-c compared to the case with DG. Thus, after installing DGs, the overall system voltage

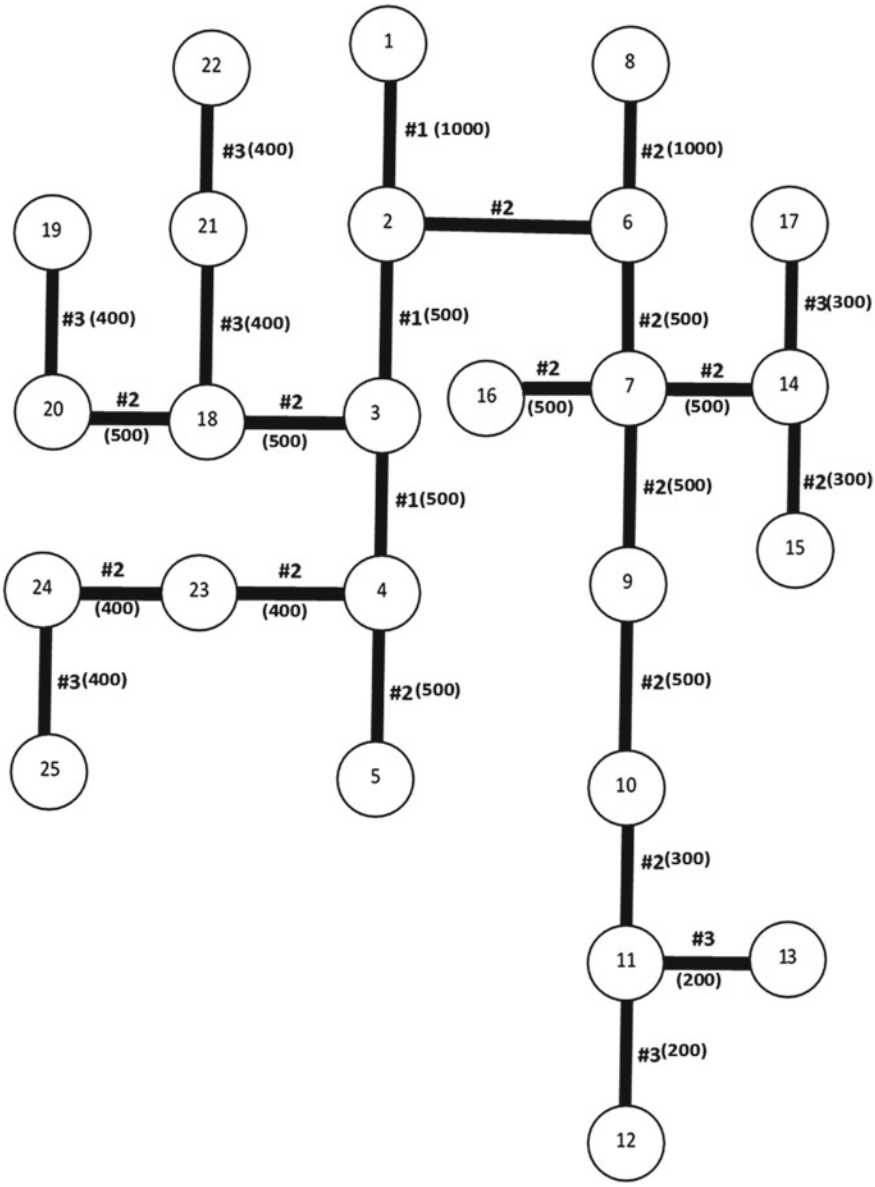


Fig. 1 Test system showing line type (#) and line length (ft.)

Table 1 Optimal location and sizing of single-phase DGs calculated by butterfly optimizer

DGs	Bus number	Phase	Real power	Reactive power
1	3	a	0.498373	0.704109
2	11	a	1.650338	0.18272
3	12	c	0.451973	0.596339
4	20	a	1.58935	1.316118
5	25	c	1.616666	0.660875

profile has improved along the phases. The maximum voltage unbalance at each bus (δV_{max}^{ph}) in case of base case is quite high as compared to one with DG case. Thus, after installing the phase-wise maximum voltage unbalance, Max(δV_{max}^{ph}) reduces considerably, i.e. from 0.027 p.u to 0.018 p.u. The system phase utilization index (PUI) is calculated using the following expression

$$PUI = \frac{Max\{|I_a - I_{avg}|, |I_b - I_{avg}|, |I_c - I_{avg}|\} \times 100}{I_{avg}} \tag{17}$$

4.1 Summary of Results

In Table 4, observed results of some important parameters during the studies are portrayed. In case of each phase, the minimum bus voltages, V_{min}^a, V_{min}^b and V_{min}^c for with and without DGs are reported in Table 4. It is seen from Table 4 that there is improvement in every parameter when DGs are considered.

When DGs are installed in the actual cases, following is perceived. The zero-sequence current ($|I_0|$), negative-sequence current ($|I_2|$) and phase utilization index (PUI) are almost negligible. The system voltages obtained before and after considering DGs are shown in Fig. 2.

5 Conclusion

It has been demonstrated that an optimal placement and sizing of single-phase DGs reduces the negative- and zero-sequence currents at main substation transformer. It is found that the bus voltages also improve due to optimal placement and sizing of DGs which consequently decreases the line losses in the system.

BO algorithm is successfully applied to obtain optimal placement and sizing of single-phase DGs in radial distribution network. It is discovered that the load at main transformer is balanced in terms of phase currents.

Table 2 Comparison of load distribution of base load without DGs and with DGs

Bus	Base load						System with DGs					
	$p_{i0}^{a,sp}$	$q_{i0}^{a,sp}$	$p_{i0}^{b,sp}$	$q_{i0}^{b,sp}$	$p_{i0}^{c,sp}$	$q_{i0}^{c,sp}$	$p_{i0}^{a,sp}$	$q_{i0}^{a,sp}$	$p_{i0}^{b,sp}$	$q_{i0}^{b,sp}$	$p_{i0}^{c,sp}$	$q_{i0}^{c,sp}$
1	9.79E+00	6.89E+00	5.87E+00	4.42E+00	8.04E+00	5.80E+00	5.88E+00	4.44E+00	5.88E+00	4.44E+00	5.88E+00	4.44E+00
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	−3.60E−01	−2.16E−01	−2.88E−01	−1.92E−01	−4.20E−01	−2.64E−01	1.38E−01	4.88E−01	−2.88E−01	−1.92E−01	−4.20E−01	−2.64E−01
4	−5.76E−01	−4.32E−01	−4.80E−02	−3.36E−02	−4.80E−01	−3.00E−01	−5.76E−01	−4.32E−01	−4.80E−02	−3.36E−02	−4.80E−01	−3.00E−01
5	−4.32E−01	−2.88E−01	−2.88E−01	−1.92E−01	−3.60E−01	−2.40E−01	−4.32E−01	−2.88E−01	−2.88E−01	−1.92E−01	−3.60E−01	−2.40E−01
6	−4.32E−01	−2.88E−01	−3.36E−01	−2.40E−01	−3.00E−01	−3.00E−01	−4.32E−01	−2.88E−01	−3.36E−01	−2.40E−01	−3.00E−01	−3.00E−01
7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8	−4.32E−01	−2.88E−01	−2.88E−01	−1.92E−01	−3.60E−02	−2.40E−02	−4.32E−01	−2.88E−01	−2.88E−01	−1.92E−01	−3.60E−02	−2.40E−02
9	−7.20E−01	−5.04E−01	−3.84E−01	−2.88E−01	−4.80E−01	−3.00E−01	−7.20E−01	−5.04E−01	−3.84E−01	−2.88E−01	−4.80E−01	−3.00E−01
10	−3.60E−01	−2.16E−01	−2.88E−01	−1.92E−01	−4.20E−01	−2.64E−01	−3.60E−01	−2.16E−01	−2.88E−01	−1.92E−01	−4.20E−01	−2.64E−01
11	−5.04E−01	−3.17E−01	−2.40E−01	−1.44E−01	−3.60E−01	−2.40E−01	1.15E+00	−1.34E−01	−2.40E−01	−1.44E−01	−3.60E−01	−2.40E−01
12	−5.76E−01	−3.60E−01	−4.80E−01	−3.36E−01	−4.80E−01	−3.60E−01	−5.76E−01	−3.60E−01	−4.80E−01	−3.36E−01	−2.80E−02	2.36E−01
13	−6.48E−01	−2.16E−01	−3.36E−01	−2.11E−01	−3.60E−01	−2.40E−01	−6.48E−01	−2.16E−01	−3.36E−01	−2.11E−01	−3.60E−01	−2.40E−01
14	−5.76E−01	−3.60E−01	−3.84E−01	−2.88E−01	−6.00E−01	−4.20E−01	−5.76E−01	−3.60E−01	−3.84E−01	−2.88E−01	−6.00E−01	−4.20E−01
15	−7.20E−02	−4.32E−02	−4.80E−02	−2.88E−02	−6.00E−02	−3.60E−02	−7.20E−02	−4.32E−02	−4.80E−02	−2.88E−02	−6.00E−02	−3.60E−02
16	−5.76E−01	−4.32E−02	−3.84E−02	−2.88E−01	−4.80E−01	−3.60E−01	−5.76E−01	−4.32E−02	−3.84E−02	−2.88E−01	−4.80E−01	−3.60E−01
17	−5.76E−01	−4.32E−01	−3.36E−01	−2.40E−01	−5.40E−01	−3.84E−01	−5.76E−01	−4.32E−01	−3.36E−01	−2.40E−01	−5.40E−01	−3.84E−01
18	−5.76E−01	−4.32E−01	−3.84E−01	−2.88E−01	−4.80E−01	−3.60E−01	−5.76E−01	−4.32E−01	−3.84E−01	−2.88E−01	−4.80E−01	−3.60E−01
19	−8.64E−02	−6.48E−02	−4.80E−02	−3.36E−02	−6.00E−02	−4.80E−02	−8.64E−02	−6.48E−02	−4.80E−02	−3.36E−02	−6.00E−02	−4.80E−02
20	−5.04E−01	−3.60E−01	−3.84E−01	−2.88E−01	−5.40E−01	−3.84E−01	1.09E+00	9.56E−01	−3.84E−01	−2.88E−01	−5.40E−01	−3.84E−01
21	−5.76E−02	−4.32E−02	−3.36E−02	−2.40E−02	−5.40E−02	−3.84E−02	−5.76E−02	−4.32E−02	−3.36E−02	−2.40E−02	−5.40E−02	−3.84E−02
22	−7.20E−01	−5.04E−01	−5.76E−01	−4.32E−01	−6.00E−01	−4.80E−01	−7.20E−01	−5.04E−01	−5.76E−01	−4.32E−01	−6.00E−01	−4.80E−01
23	−8.64E−02	−6.48E−01	−4.80E−02	−3.84E−02	−6.00E−01	−4.20E−01	−8.64E−02	−6.48E−01	−4.80E−02	−3.84E−02	−6.00E−01	−4.20E−01
24	−5.04E−01	−3.60E−01	−4.32E−01	−3.07E−01	−4.80E−02	−3.60E−02	−5.04E−01	−3.60E−01	−4.32E−01	−3.07E−01	−4.80E−02	−3.60E−02
25	−8.64E−02	−6.48E−02	−4.80E−02	−2.88E−02	−6.00E−02	−4.20E−02	−8.64E−02	−6.48E−02	−4.80E−02	−2.88E−02	1.56E+00	6.19E−01

Table 3 Comparison of voltages after and before re-phasing for constant load model

Bus	Base case				After re-phasing			
	$V^a(Ang)$	$V^b(Ang)$	$V^c(Ang)$	δV_{max}^{ph}	$V^a(Ang)$	$V^b(Ang)$	$V^c(Ang)$	δV_{max}^{ph}
1	1.05 (0.00)	1.05 (−120.00)	1.05 (120.00)	0.00	1.05 (0.00)	1.05 (−120.00)	1.05 (120.00)	0.00
2	1.03 (−0.59)	1.04 (−120.07)	1.03 (119.54)	0.01	1.04 (−0.26)	1.04 (−120.14)	1.04 (119.63)	0.00
3	1.02 (−0.69)	1.03 (−120.11)	1.03 (119.45)	0.01	1.03 (−0.30)	1.03 (−120.19)	1.04 (119.58)	0.00
4	1.02 (−0.72)	1.03 (−120.12)	1.03 (119.41)	0.02	1.03 (−0.34)	1.03 (−120.22)	1.04 (119.62)	0.01
5	1.02 (−0.73)	1.03 (−120.12)	1.03 (119.41)	0.02	1.03 (−0.34)	1.03 (−120.22)	1.03 (119.61)	0.01
6	1.01 (−0.70)	1.03 (−120.01)	1.02 (119.53)	0.02	1.03 (−0.26)	1.03 (−120.10)	1.03 (119.59)	0.00
7	1.00 (−0.81)	1.02 (−119.95)	1.01 (119.51)	0.02	1.02 (−0.25)	1.02 (−120.06)	1.02 (119.53)	0.00
8	1.01 (−0.71)	1.03 (−120.03)	1.02 (119.54)	0.02	1.02 (−0.27)	1.03 (−120.11)	1.03 (119.59)	0.01
9	1.00 (−0.87)	1.02 (−119.94)	1.01 (119.50)	0.02	1.01 (−0.18)	1.02 (−120.07)	1.02 (119.49)	0.01
10	0.99 (−0.92)	1.02 (−119.93)	1.00 (119.49)	0.03	1.01 (−0.11)	1.01 (−120.09)	1.02 (119.45)	0.01
11	0.99 (−0.95)	1.02 (−119.93)	1.00 (119.49)	0.03	1.01 (−0.07)	1.01 (−120.10)	1.02 (119.43)	0.01
12	0.99 (−0.95)	1.02 (−119.92)	1.00 (119.49)	0.03	1.01 (−0.07)	1.01 (−120.10)	1.02 (119.41)	0.01
13	0.99 (−0.96)	1.02 (−119.93)	1.00 (119.49)	0.03	1.01 (−0.08)	1.01 (−120.10)	1.02 (119.43)	0.01
14	1.00 (−0.82)	1.02 (−119.94)	1.01 (119.50)	0.02	1.02 (−0.26)	1.02 (−120.05)	1.02 (119.52)	0.00
15	1.00 (−0.82)	1.02 (−119.94)	1.01 (119.50)	0.02	1.02 (−0.26)	1.02 (−120.05)	1.02 (119.52)	0.00
16	1.00 (−0.85)	1.02 (−119.91)	1.01 (119.50)	0.02	1.02 (−0.29)	1.02 (−120.02)	1.02 (119.53)	0.00
17	1.00 (−0.82)	1.02 (−119.93)	1.01 (119.50)	0.02	1.01 (−0.26)	1.02 (−120.05)	1.02 (119.53)	0.01
18	1.02 (−0.70)	1.03 (−120.09)	1.02 (119.45)	0.02	1.03 (−0.30)	1.03 (−120.16)	1.03 (119.56)	0.00
19	1.01 (−0.70)	1.03 (−120.08)	1.02 (119.45)	0.02	1.03 (−0.30)	1.03 (−120.14)	1.03 (119.53)	0.01
20	1.01 (−0.70)	1.03 (−120.08)	1.02 (119.45)	0.02	1.04 (−0.30)	1.03 (−120.14)	1.03 (119.53)	0.01
21	1.01 (−0.70)	1.03 (−120.08)	1.02 (119.46)	0.02	1.03 (−0.30)	1.03 (−120.15)	1.03 (119.57)	0.00
22	1.01 (−0.70)	1.03 (−120.08)	1.02 (119.47)	0.02	1.03 (−0.30)	1.02 (−120.15)	1.03 (119.58)	0.00
23	1.02 (−0.69)	1.03 (−120.12)	1.02 (119.41)	0.02	1.03 (−0.30)	1.03 (−120.25)	1.04 (119.67)	0.01
24	1.01 (−0.69)	1.03 (−120.13)	1.02 (119.41)	0.02	1.03 (−0.30)	1.03 (−120.28)	1.04 (119.73)	0.01
25	1.01 (−0.69)	1.03 (−120.13)	1.02 (119.42)	0.02	1.03 (−0.30)	1.03 (−120.29)	1.04 (119.80)	0.02
	$Min(V^a)$	$Min(V^b)$	$Min(V^c)$	Max(δV_{max}^{ph})	$Min(V^a)$	$Min(V^b)$	$Min(V^c)$	Max(δV_{max}^{ph})
	0.99	1.02	1.00	0.03	1.01	1.01	1.02	0.02

Table 4 Summary of results of important quantities

Parameter	Without *DGs*	With *DGs*
V^a_{min}	0.98866	1.01088
V^b_{min}	1.01502	1.01130
V^c_{min}	1.00171	1.01659
P_{loss}	0.028084	0.017326
Q_{loss}	0.043635	0.025307
$\|I_0\|$	1.274243	4.37E−05
$\|I_2\|$	1.280872	1.88E−05
PUI	0.246035	7.2E−06

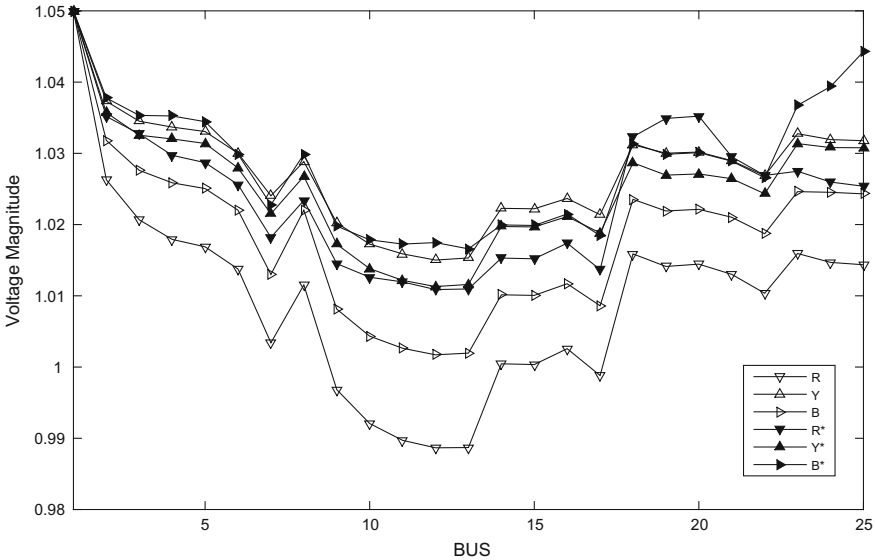


Fig. 2 System voltages with and without DGs

References

1. Mishra, M.K., Joshi, A., Ghosh, A.: A new compensation algorithm for balanced and unbalanced distribution systems using generalized instantaneous reactive power theory. Electr. Power Syst. Res. **60**, 29–37 (2001)
2. Araujo, L.R., Penido, D.R.R., Carneiro, S., Pereira, J.L.R.: A three-phase optimal power-flow algorithm to mitigate voltage unbalance. IEEE Trans. Power Deliv. **28**(4), 2394–2402 (2013)
3. Siti, W.M., Jimoh, A., Nicolae, D.: Distribution network phase load balancing as a combinatorial optimization problem using fuzzy logic and Newton-Raphson. Electr. Power Syst. Res. **81**, 1079–1087 (2011)
4. Kumar, A., Misra, R.K., Singh, D.: Butterfly optimizer. In: IEEE Workshop on Computational Intelligence, Theories, Applications and Future Directions (WCI), pp. 1–6. Kanpur (2015)

5. Lin, W.-M., Su, Y.-S., Chin, H.-C., Teng, J.-H.: Three-phase unbalanced distribution power flow solutions with minimum data preparation. IEEE Trans. Power Syst. **14**(3), 1178–1183 (1999)
6. Viswanadha Raju, G.K., Bijwe, P.R.: Efficient reconfiguration of balanced and unbalanced distribution systems for loss minimisation. IET Gener. Transm. Distrib. **2**(1), 7–12 (2008)
7. Cheng, C.S., Shirmohammadi, D.: A three-phase power flow method for real-time distribution system analysis. IEEE Trans. Power Syst. **10**(2), 671–679 (1995)

Differential Evolution-Based Matched Wavelet for Differential Protection of Transformer

Arpita Roy, Devender Singh and Rakesh K. Misra

Abstract The work proposes a matched wavelet method for detection and discrimination of inrush and fault waveform in power transformers. The method rests on the concept that if analyzing wavelet is specialized to have shape matching with the waveform being analyzed (fault or inrush) its discrimination ability can improve drastically. The matched wavelet for inrush and fault waveforms is developed using Differential Evolution (DE) algorithm. It is established that the proposed method has sub-cycle (nearly half-cycle) discrimination ability. The method was tested by generating waveforms for all the angles to test its discrimination ability.

Keywords Harmonic restraint · Matched wavelet · DE

1 Introduction

Transformers are the power systems' vital component, and their protection is the main challenge in the electrical domain. There are many ways to achieve this purpose, in which differential relays is the primary one. It works on the principle of taking the difference between the primary and secondary current of transformer. The known limitation of differential relays is its inability to differentiate between fault current and magnetizing inrush. During the sequence of initial activation of transformer, there is a sudden rise in the current, termed as inrush which may be misinterpreted as fault by differential relays [1].

The second harmonic component of differential current is quite large as compared to its fundamental component. This difference has been used as basis for detection

A. Roy (✉) · D. Singh · R. K. Misra
IIT (BHU) Varanasi, Varanasi 221005, Uttar Pradesh, India
e-mail: aroy.rs.eee13@iitbhu.ac.in

D. Singh
e-mail: dsingh.eee@iitbhu.ac.in

R. K. Misra
e-mail: rkmisra.eee@iitbhu.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_40

of inrush current by prevalent differential relays in harmonic restraint method. Pros of second harmonic restraint method weigh out its cons and thus most widely used method in comparison to other existing ones like neural network [2, 3], fuzzy Logic [4–6], wavelet transform [7], and adaptive relay.

Chapa has introduced a new concept for representing any arbitrary signal as matched wavelet [8]. Wavelet is a mathematical tool capable of analyzing a given signal into varied scale component, having their own frequency range at each scale [9]. The method for designing wavelet matching to its source signal involves two underlying sets of equations for amplitude and phase representation. This representation of signal in terms of matched wavelet assists in pattern recognition.

This paper is about obtaining the wavelet similar in shape to inrush and fault waveform of transformer by using Differential Evolution optimizing algorithm. In further sections, we have elaborated the DE algorithm and tested the effectiveness of matched wavelet for the detection of inrush and fault current. The obtained result is discussed and listed in tables for all possible parameters.

2 DE Algorithm

The cost function is generally nonlinear, non-continuous, and non-differential in case of many practical optimization problems. Algorithms such as differential evolution, evolutionary strategies, and genetic algorithms are useful for such optimization problems. DE is simple.

DE has low space complexity or it requires lesser memory while implementation as compared to other real parameter optimizers. DE can thus handle large problems of optimization.

DE works as a population-based technique which exploits NP variables as population of D-dimension parameter vector for each generation. In general, DE produces new solutions using summation of scaled difference vector of two individual of population to a third individual. Further if the subsequent new solution leads to a better value of cost function than older individual, then the lately generated solution replaces the current individual.

The details of the DE processes are described in subsequent sections.

2.1 Initialization

DE starts with a randomly generated initial population of N_p, D-dimensional real-valued individuals (solution vectors). Hence, ith individual for current generation is denoted by:

$$\mathbf{X}_{i,t} = [\mathbf{x}_{1,i,t}, \mathbf{x}_{2,i,t}, \mathbf{x}_{3,i,t}, \ \ldots \ , \mathbf{x}_{D,i,t}.] \tag{1}$$

Randomly generated initial population is expected to uniformly cover the whole search space, and it is virtually possible by uniformly randomizing individuals in the search space. Hence, jth element of the ith individual can be initialized by

$$\mathbf{x}_{i,j,t=0} = \mathbf{x}_{i,j\ min} + \mathbf{rnd} \cdot (\mathbf{x}_{i,j\ max} - \mathbf{x}_{i,j\ min}) \tag{2}$$

where **rnd** represents the uniformly distributed random number in the range (0, 1].

2.2 Mutation

The mutation operation in DE algorithm results in mutant vector $XV_{i,t+1}$. The mutation operation is performed using three parameter vectors taken from the current population. These three participating parameter vectors are called as target vector ($X_{s,t}$) and donor vectors ($X_{r,t}$ and $X_{q,t}$). The mutant vector is obtained using following operations on these three vectors.

$$XV_{i,t+1} = X_{s,t} + sF(X_{r,t} - X_{q,t}) \tag{3}$$

where random indexes $s, r, q \in \{1, 2, 3 \ldots N_P\}$ and running indexes i are mutually exclusive. $sF \geq 0$ is a scaling factor $\in [0, 2]$ which scales the difference vector $(X_{r,t} - X_{q,t})$.

2.3 Crossover

The mutant vector obtained after mutation operation undergoes a crossover operation. The crossover is executed using a target vector $X_{i,t}$ and mutant vector $XV_{i,t+1}$. A new parameter vector called a trial vector, $Y_{i,t} = [y_{1,i,t}, y_{2,i,t}, y_{3,i,t} \ldots, y_{D,i,t}]$, is obtained after the crossover operation. The elements, $y_{j,i,t+1}$, of the trial vector, $Y_{i,t+1}$, are obtained from the target vector, $X_{i,t}$, and mutant vector, $XV_{i,t+1}$, through process defined by following equation.

$$y_{j,i,t+1} = \begin{cases} xv_{j,i,t+1} & (rb(j) \leq cr)\ or\ (j = rbr(i)) \\ x_{j,i,t} & (rb(j) > cr)\ and\ (j \neq rbr(i)) \end{cases} \tag{4}$$

where $rb(j)$ is a uniform random number generator from [0, 1], cr is the crossover probability, $rbr(i)$ is a randomly selected index from $\in$ 1,2,… D which is used to ensure that at least one element from $xv_{j,i,t+1}$ is selected in $y_{j,i,t+1}$. In the scheme of crossover procedure, $X_{i,t}$ is the target vector and $XV_{i,t+1}$ is the mutant vector. Trial vector $Y_{i,t+1}$ is initially equal to $X_{i,t}$, and as the crossover process is performed, the elements of mutant vector $XV_{i,t+1}$ replace the elements of $X_{i,t}$.

2.4 Selection

The final step of DE is selection in which it is ensured that the trial or target vector sustains to the future generations. The selection procedure is written mathematically as:

$$X_{i,t} = \begin{cases} Y_{i,t+1} & f(Y_{i,t+1}) \leq f(X_{i,t}) \\ X_{i,t} & f(Y_{i,t+1}) > f(X_{i,t}) \end{cases} \tag{5}$$

where $f(\mathbf{X})$ is the cost function to be minimized. Now, if the newly obtained trial vector returns a best or equal value of the cost, it supersedes the existing target individual for next generation.

2.5 Gradient Repair Method

The key concern in evolutionary methods is to check whether the obtained optimized solution is satisfying the set of constraints or not. If the solution satisfies all the constraints then the solution is said to lie in feasible region, but if not then there is constraint violation and hence feasible solution must be checked and calculated. This gradient repair method is used for finding feasible solution.

Following are the steps for gradient repair method:

1. Let W consist of vectors of inequality constraints (g) and equality constraints (h) for the problem.
$$W = \begin{bmatrix} g_{M\times 1} \\ h_{N\times 1} \end{bmatrix}.$$
2. The gradient, ∇W, gives by what extent the existing solution is defied for given constraints.
3. $\nabla_S W$ is determined in reference to solution point, $S = \{s_i \mid \forall i = 1, 2, \ldots D\}$, and can be evaluated as $\nabla_S W = \begin{bmatrix} \nabla_S g \\ \nabla_S h \end{bmatrix}$. $\nabla_S W$ can be approximated by use of the following forward difference formula.
$$\nabla_S W = \begin{bmatrix} \frac{g(S|s_i = s_i+\epsilon) - g(S|s_i = s_i))\ \forall i = 1,2,\ldots D}{\epsilon} \\ \frac{h(S|s_i = s_i+\epsilon) - h(S|s_i = s_i))\ \forall i = 1,2,\ldots D}{\epsilon} \end{bmatrix}$$
where ϵ is a small disruption for computing the gradients. The extent of constraint violation corresponding to all the variables can be written as $\Delta W = \nabla_S W \times \Delta S$. Following relation can be used for repairing the solution.
$$\Delta S = \nabla_S W^{-1} \times \Delta W$$

4. The amount of constraint violation, ΔW, is resolved using the upper bound vector, g_u, and the lower bound vector, g_l, of inequality constraints, g, and right-hand side equality constraints, c, using the following relation.

$$\Delta W = \begin{bmatrix} Min\{0, g_u - g(S)\} + Max\{0, g_l - g(S)\} \\ h(S) - c \end{bmatrix}.$$

In general, $\nabla_S W$ is not a square matrix; therefore, its pseudo-inverse, $\nabla_S W^+$, is used for rectifying the solution, S^{old}, by using the following formula.

$$S^{new} = S^{old} + \nabla_S W^+ \times \Delta W$$

5. The constraints are evaluated for this new solution, and this is done repetitively until the feasible solution is obtained.

3 Application of DE Algorithm in Obtaining Matched Wavelet for Inrush and Fault Waveform

A mother wavelet in general can be defined as

$$\begin{aligned} \psi(x) &= \sum h(n)\phi(2x - n)\ where, \\ h(k) &= (-1^k) l(N - K - 1) \end{aligned} \tag{6}$$

$h(k)$ behaves like high-pass filter coefficient, $l(k)$ behaves like low-pass filter coefficient, and $\phi(x)$ is called scaling function. So as per the present job, a matched wavelet for a inrush waveform similar to waveform in shape is to be determined. Thus, objective function can be expressed as $F(x) = \sum[f(x_i) - \psi(x_i)]^2\ for i = 1, 2, \ldots n$, in which the function $f(x_i)$ are the data points which are to be matched in least squares sense. The various parameters of wavelet function,$\psi(x)$, are number of coefficients, $h(k)$, and number of constraints. Various constraints on the wavelet coefficients are:

$$l(0) + l(1) + l(2) + \cdots + l(N) = \sqrt{2} \tag{7}$$

$$\sum_0^q l^2(q) = 1, \quad q = 0, 1, 2 \ldots N - 1 \quad N \in Z \tag{8}$$

$$\begin{aligned} \sum_0^q l(l)l(q - 2k) = \sum_0^q l(q)l(q + 2k) = 0 \\ q = 0, 1, 2 \ldots N - 1, \ for\ \ all\ k \neq 0 \end{aligned} \tag{9}$$

$$\sum_{k}^{N}(-1)^k k^p l(k) = 0 \; where \; p = degree \; of \; moment \tag{10}$$

Equation (7) gives the condition that the sum of all wavelet coefficients must be equal to 1. The second constraint, (8), results from the fact that integer translates of scaling function are orthonormal to each other. The condition given by (9), called as double shift orthogonality, ensures that sum of product of doubly shifted wavelet coefficients is equal to zero.

With the help of DE algorithm, the objective function which is least mean square error between the matched wavelet and inrush waveform can be minimized and the best possible matched inrush wavelet was found for a given number of coefficient and constraints. Different numbers of coefficients were investigated for generating matched wavelet. The numbers of coefficients for which wavelets are designed are 8, 12, 16. The number of coefficient corresponds to the dimension of population vector (D), in DE.

The values for the various DE parameter used in this study are F = 0.7, crossover constant CR = 0.9, and number of population NP is 80 for dimension $D < 10$ and $NP = 400$ for dimension greater than or equal to 10.

3.1 Time and Frequency Domain Analysis

The approach of finding matched wavelet is performed in two domains, namely time and frequency domain. For the reason of having lesser number of wavelet coefficients in frequency domain, the proposed method is tested for both time and frequency domain. The inrush or fault waveform $f(x)$ in time domain is expressed as, $F = \sum(f(x) - \psi(x))^2$. The equation in frequency domain is expressed as, $F = \sum(f(\omega) - \psi(\omega))^2$.

4 Results and Discussions

Three-phase power transformer with 25 MVA power ratings, 150/75 KV winding voltages, saturable core, Y-Y connection at 60 Hz is used in this work, and modeling of transformer is not discussed due to space limitations. The above transformer model is used for getting the various waveforms presenting fault and inrush occurrence. The waveforms are obtained at different angles from 0° to 180° at the interval of 5°.

Although the matched wavelet is designed for various number of coefficients, best among them is decided on the basis of its efficiency to detect. At the same time, lower number of filter coefficients are preferred for implementation on hardware.

Inrush waveform per cycle of power frequency and matched wavelet for 12 coefficients is shown in Fig. 1. From the figure, it can be seen that the number of samples

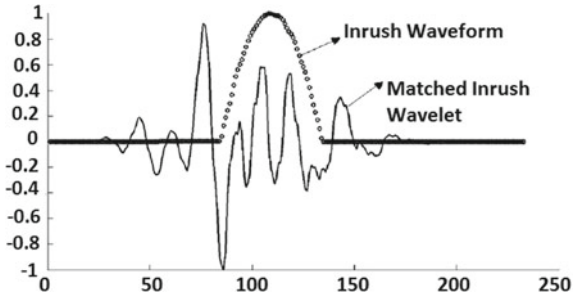


Fig. 1 Matched wavelet for inrush waveform

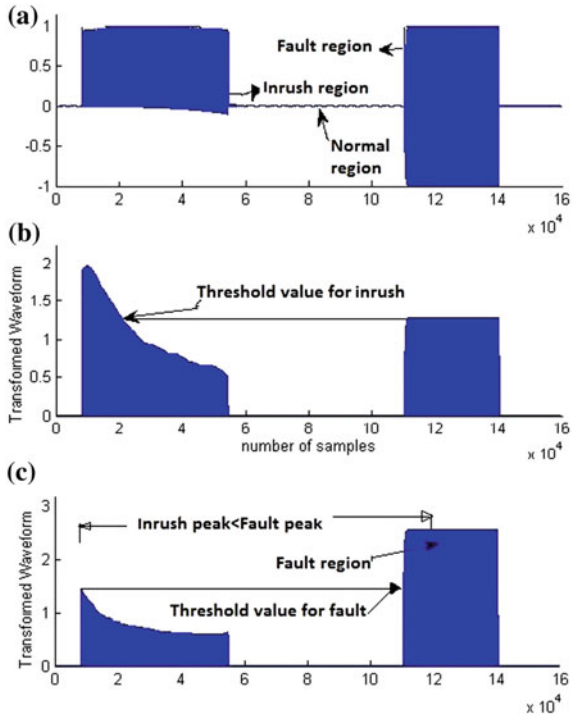


Fig. 2 **a** Original waveform at 0°. **b** Transformed waveform using inrush wavelet at 0°. **c** Transformed waveform using fault wavelet at 0°

of inrush current depends on the number of wavelet coefficient chosen. Figure 2a shows the inrush and fault region of transformer waveforms in which the peak values are scaled to value 1. One more region is the normal region where differential current was below a preset value. In the same figure, part (b) and part (c) represent the inrush and fault waveform in transformed domain, i.e., after matched wavelet analysis. So matched inrush wavelet gives high peak value at the inrush region of waveform as shown in Fig. 2b. Also, the fault matched wavelet gives high peak value in the fault region of transformer waveform as shown in Fig. 2c. The discrimination for the occurrence of inrush and fault is done with the help of a threshold value which is the value between the two peaks in the transformed waveforms. This threshold

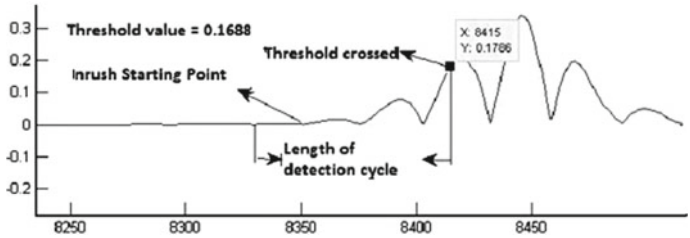


Fig. 3 Enlarged view transformed waveform

value is evaluated in all cases of different switching angles. Then the maximum value among them is chosen as final threshold value to differentiate inrush from fault. This procedure is done for both inrush and fault matched wavelets.

Figure 3 explains calculating the number of cycles involved in detection of inrush current. From Fig. 3, it can be noticed that starting sample number of inrush is 8350, and as per the decided threshold (0.1688), the sample number at which it is crossed is 8415, so 65 samples are required to detect inrush pattern in waveform. For calculating the number of cycles involved in detection, the number of samples required for crossing threshold is divided by the total number of samples per cycle of waveform. Similarly for the fault waveform, the threshold value is obtained. Figure 2c shows that fault matched wavelet gives higher peak value in the fault region compared to the inrush region in the transformed waveform.

Table 1 shows threshold values at different switching angles for inrush waveform. Column 1 represents switching angles; in column 2, threshold value is demonstrated;

Table 1 Inrush and fault analysis of a three-phase transformer using matched wavelet at various angles

Inrush			Fault	
Coefficient = 12		Constant = 4	Coefficient = 16	Constant = 6
Angle	Threshold	Cycle for max. threshold	Threshold	Cycle for max. threshold
0	0.1688	0.1021	1.4545	0.3994
25	0.1688	0.0991	1.3495	0.3363
50	0.1688	0.1471	0.8886	0.2943
75	no inrush	no inrush	no inrush	no inrush
100	–	–	–	–
125	0.1688	0.1411	0.8407	**0.5676**
145	0.1688	**0.3904**	1.2765	0.5075
150	0.1688	0.0961	1.3276	0.4895
175	0.1688	0.1021	1.4483	0.4114
180	0.1688	1.1021	1.4552	0.3994

and cycles required for detection of inrush current are indicated in column 3 of Table 1. The highest number of cycle required is 0.3904 as highlighted in Table 1, which means that inrush can be detected in at least 0.3904 cycles. The column 2 and column 3 of this table denote values regarding inrush wavelet designed with 12 coefficients and 14 constraints in time domain.

The column 4 and column 5 of the same table represent data with reference to the fault wavelet. The maximum value of threshold in this case is 1.4552, and the number of cycles required to reach the maximum threshold value corresponding to this is 0.5676, which is highlighted in column 5 of the same table. Hence, this means if the matched wavelet with 16 coefficient and 6 constraints is designed, then it can be detected by at most 0.5676 cycles.

As explained in Sect. 3, with the idea of getting better matching in least square sense, matched wavelet is designed in the frequency domain also. In this case, all calculations are done on Fourier transform of the fault waveform. It may be noted that the number of coefficients in case of frequency domain are 8 and 12, respectively, for inrush and fault analysis which are less than corresponding number of coefficients (12 and 16, respectively) in time domain as shown in Table 2. In case of frequency domain, the minimum number of cycles required for detection of fault and inrush is indicated in Table 2. Column 1 shows the switching angle, and its respective threshold value is shown in column 2 for the inrush waveform. The maximum threshold value is 0.4480. The number of cycles required to reach the maximum threshold value corresponding to switching angle is 0.5255 cycles.

In columns 4 and 5 of Table 2, respective threshold value and number of cycles for fault waveform in frequency domain are shown. The maximum value of threshold among all switching angle is 1.4009, and it is highlighted in column 4. The number of

Table 2 Inrush and fault analysis of a three-phase transformer using matched wavelet in frequency domain at various angles

Inrush			Fault	
Coefficient = 8		Constant = 10	Coefficient = 12	Constant = 7
Angle	Threshold	Cycle for max. threshold	Threshold	Cycle for max. threshold
0	0.448	0.1321	1.4009	0.7718
25	0.448	0.1351	1.21	0.3844
50	0.448	0.1351	0.8304	0.3604
75	no inrush	no inrush	no inrush	no inrush
100	no inrush	no inrush	no inrush	no inrush
125	0.448	0.5105	0.7461	0.7417
135	0.448	**0.5255**	1.0107	0.7057
150	0.448	0.1321	1.183	**0.8619**
175	0.448	0.1321	1.3913	0.7838
180	0.448	0.1321	1.4552	0.7688

cycles required to reach the maximum threshold value is 0.8619 as shown in column 5 which declares that the fault waveform can be detected in at least 0.8619 cycles. Impact of noise has not been performed in this present work.

5 Conclusion

In this paper, a matched wavelet method for detection and discrimination of inrush and fault waveform in power transformers. It is demonstrated that by using matched wavelet concept, detection and discrimination of inrush and fault waveform can improve drastically. It is also demonstrated that there are several optimal matched wavelets possible depending on the number of coefficients and smoothness constraints employed. In this work, the matched wavelets for inrush and fault wavelets are developed using Differential Evolution algorithm. An elaborated testing mechanism was adopted to get the worst-case time for discrimination by using all the different fault and inrush angles. Also, the matched wavelet was developed only for the waveform at zero degrees and it was used for discrimination of fault and inrush for waveform generated at all the different angle. It is established that the present method has sub-cycle (nearly half-cycle) discrimination ability.

References

1. Yabe, K.: Power differential method for discrimination between fault and magnetizing inrush current in transformer.IEEE Trans. Power Syst. Delivery **12**(3), 1109–1118 (1997)
2. Segatto, E.C., Coury, D.V.: A differential relay for power transformers using intelligent tools. Power Syst. IEEE Trans. **21**, 1154–1162 (2006)
3. Tripathy, M., Maheshwari, R.P., Verma, H.K.: Power transformer differential protection based on optimal probabilistic neural network. IEEE Trans. Power Delivery **25**(1) (2010)
4. Girgis, A.A., Hart, D.G., Chang, W.B.: An adaptive scheme for digital protection of power transformers. IEEE Trans. Power Delivery **7**(2), 546–553 (1992)
5. Shin, M., Park, C., Kim, J.K.: Fuzzy logic based relaying for large power transformer protection. IEEE Trans. Power Delivery **18**(3), 718–723 (2003)
6. Wiszniewski, A., Kasztenny, B.: A multi-criteria differential transformer relay based on fuzzy logic. IEEE Trans. Power Delivery **10**(4), 1786–1792 (1995)
7. Bhasker, S.K., Tripathy, M., Kumar, V.: Wavelet transform based discrimination between inrush and internal fault of indirect symmetrical phase shift transformer. In: IEEE PES General Meeting|Conference and Amp, pp. 1–5 (2014)
8. Chapa, J.O., Rao, R.M.: Algorithm for designing wavelets to match a specified signal. IEEE Trans. Signal Process. 3395–3406 (2000)
9. Graps, A.: An introduction to wavelets. IEEE Trans. Comput. Sci. Eng. 50–61 (1995)
10. Habib, M., Marin, M.A.: A comparative analysis of digital relaying algorithms for the differential protection of three phase transformers. IEEE Trans. Power Syst. **3**, 1501–1508 (1988)
11. Rahman, M.A., Jeyasurya, B.: A state-of-the-art review of transformer protection algorithms. IEEE Trans. Power Delivery **3**, 534–544 (1988)

12. Gomez-Morante, M., Denise, W.: A wavelet-based differential transformer protection. IEEE Trans. Power Delivery **14**(4) (1999)
13. Tewfik, A.H., Sinha, D., Jorgensen, P.: On the optimal choice of a wavelet for signal representation. IEEE Trans. Inform. Theory **38**, 747–765 (1992)
14. Chapa, J.O.: Matched wavelet construction and its application to target detection. Ph.D. Dissertation, Rochester Inst. Technol, Rochester, NY, Aug 1995

Part VII
Pattern Recognition and Medical Imaging

Dynamic Texture Recognition from Multi-offset Temporal Intensity Co-occurrence Matrices with Local Pattern Matching

Seba Susan, Minni Mittal, Srishti Bansal and Prachi Agrawal

Abstract In this paper, we propose dynamic texture recognition from video snippets by constructing temporal intensity co-occurrence histograms for feature representation and learning. The pair-wise intensity co-occurrence frequencies are summarized from every pixel position between every pair of sequential frames in the video separated by a certain time lapse or offset distance. A 256×256 grayscale intensity co-occurrence matrix is thus constructed for the given offset distance. Twenty offset distances from $d = 1, 2, \ldots, 20$ are used for the computation that yields twenty 256×256 temporal co-occurrence matrices from a single video. The twenty 2D histograms so formed are individually converted to complete probability distributions whose elements sum up to one, and then each histogram is converted to a 1D feature vector. The twenty 1D feature vectors represent local patterns that are concatenated to form a unique ID pattern that is matched using the ensemble of bagged decision trees classifier. Alternatively, a 20×20 grid of distance classifiers is substituted to find matches between the local patterns followed by the summation of distances from all the grids. The recognition rate achieved in our experiments is found superior to the state-of-the-art, when tested on the raw, unprocessed, and unsegmented videos of the benchmark Dyntex++ dataset.

Keywords Dynamic texture recognition · Temporal co-occurrence matrix Multi-offset · Distance classifier

S. Susan (✉) · M. Mittal · S. Bansal · P. Agrawal
Department of Information Technology, Delhi Technological University, Bawana Road, New Delhi 110042, India
e-mail: seba_406@yahoo.in

M. Mittal
e-mail: minnimittal31@gmail.com

S. Bansal
e-mail: srishtib22@gmail.com

P. Agrawal
e-mail: prachi2311.agrawal@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_41

1 Introduction

Dynamic textures are defined as the spatiotemporal periodic patterns contained in a sequence of video frames that uniquely characterize a specific type of motion in the context of the scene involved. One direct application of dynamic texture analysis is the categorization of the real-world dynamic textures such as flowing water, sea waves, flickering flame, and fluttering flags [1]. More interestingly, the concepts and solutions of dynamic texture analysis could be applied to the modern-day video surveillance problems of human action recognition [2], facial expression analysis [3], and face recognition from videos [4, 5]. Both the visual picture and temporal dynamics are of importance when categorizing dynamic textures [6]. The categorization of dynamic textures is a challenging pattern recognition problem due to the dynamic and evolving nature of the spatiotemporal patterns. A typical solution is the development of novel spatiotemporal features dedicated for the purpose, several instances of which are found in the literature [3, 7–9]. The local binary patterns along three orthogonal planes (LBP-TOP) [3], spatiotemporal regularity features based on the normal flow field [7] and the dynamic pooled complementary features (DPCF) [8] that analyzes a video with respect to its spatial, temporal, and color properties are some of the examples. More recently, graph-theoretic solutions were proposed such as the spatiotemporal directional number transitional graph (ST-DNG) which is a micro-texture spatiotemporal descriptor extracted from local neighborhoods in the image sequence [9]. Doretto et al. devised a novel procedure of representing dynamic texture image sequences as the output of stochastic dynamic models [10]. Classification is achieved by computing the Martin distance between the stochastic dynamic models. Nonparametric motion models of the image sequences in dynamic textures are constructed in [11] from temporal and multi-scale co-occurrence statistics. The maximum likelihood criterion determined the class label in this case. Learning the sequential and periodic information in dynamic textures through improved machine learning algorithms is another aspect of research in this field. In [12], spatial and temporal features are computed separately from the 3D volume of a dynamic texture video and integrated through ensemble support vector machines (SVMs). Consequent to the surge in deep learning techniques in 2014, some works have advocated deep learning for the dynamic texture recognition problem as well [1, 2]. Deep image features were extracted from spatial and temporal convolutional neural networks (CNNs) for dynamic texture recognition in [1]. The 2D video frames with the mean subtracted were the input for the spatial CNN, and the differences between adjacent frames defined the input sequence for the temporal CNN. Multi-dimensional or more specifically 3D CNNs were proposed in [2] for recognizing the dynamic textures found in human action sequences, which is a specific problem of dynamic texture recognition. Since video blocks are of three dimensions, the 3D CNNs are apt for direct learning of the three-dimensional data points in the video block. All deep learning techniques require a lot of training to achieve high accuracies that requires the segmentation of dynamic texture datasets containing limited videos into a large number of small video snippets.

In our work, we focus on using the actual raw videos in the dataset for the experimentation without any dissection of the video into smaller snippets, which is a tougher form of recognition experiment since the training and test data are disjoint. For feature extraction, we construct the temporal co-occurrence matrix for different frame offset distances from an intensity-to-intensity mapping between every two sequential frames separated by that offset distance. The probability density matrix so formed for each offset distance is converted to a one-dimensional feature vector and applied for classification. Local pattern matching for multiple offset distances is also explored. The organization of this paper is as follows. The feature extraction stage involving the construction of temporal co-occurrence matrices is described in Sect. 2. The proposed dynamic texture recognition methodology is discussed in Sect. 3. The results are analyzed in Sect. 4, and the conclusions are drawn in Sect. 5.

2 Feature Extraction from the Temporal Co-occurrence Matrix

The co-occurrence matrix is a statistical tool for 2D texture measurement introduced by Haralick and Shanmugam [13, 14]. The regularity of texture patterns is interpreted as the repetitive co-occurrence of certain combinations of image intensities that are separated by a predetermined distance called the offset distance. The co-occurrence matrix is defined as the histogram of co-occurring intensities that are separated in the 2D (x, y) space by an offset distance of $(\Delta x, \Delta y)$. The 2D histogram is converted to a probability distribution by dividing each element of the matrix by the total sum of elements in the matrix. Statistical features such as entropy, energy, coarseness are then computed from the 2D probability co-occurrence matrix which forms the feature vector for learning. Recent improvisations related to co-occurrence-based statistics include the non-extensive entropy with Gaussian gain [15], which was proposed as a single feature for identifying regular spatial textures. This measure was summarized from a co-occurrence matrix with the offset distance equal to a quarter of the size of the texture image that recorded the frequency of pair-wise occurrences of intensities in the spatial domain. The idea was extended successfully in [16] to three-dimensional color planes, for color texture recognition. Constructing co-occurrence matrix along the time axis for detecting periodic temporal patterns in video sequences is an obvious next line of thought and is the one that is pursued in this paper.

Anomalies in the regular motion of crowds in videos are indicated by a disturbance in the continuity of the pattern of motion vectors [17]. On the other hand, periodic motion along the temporal plane is characterized by a continuity or repetition of certain combination of intensities along the time axis. One of the first pioneering works in this direction is that of Nelson and Polana in 1992 [18], in which spatial and temporal statistical motion features are extracted from the co-occurrences of normal flow fields of a video sequence. Co-occurrence matrices are used for determining

the position, scale, and rotation of moving objects in a scene in [19]. The relative position of the object in the scene is analyzed using co-occurrence matrices in order to determine the size of the object. Temporal co-occurrence matrices are constructed for a video sequence from quantified motion quantities for specific offset distances in [20]. Motion features are then computed from the co-occurrence matrix.

We first describe the formation of a single 2D co-occurrence matrix for a given offset distance from a sequence of video frames. In our experiments, we construct a 256×256 co-occurrence matrix along the time axis, where each cell is the frequency of occurrence of a pair of intensities along the time axis at the same pixel position separated by an offset distance $offd = \Delta t$. The matrix cells are initialized to zero at the beginning. Consider a video sequence $\mathbf{z}$ containing *nFrames* number of frames where each pixel position in the tth frame is denoted by the vector (i, j). Then, the (m, n) cell of the 256×256 temporal co-occurrence matrix for a predefined frame offset distance *offd* is updated as

$$C_{offd}(m, n) = C_{offd}(m, n) + 1 \big| \mathbf{z}(i, j, t) = m, \mathbf{z}(i, j, t + offd) = n, \forall t = 1, 2, \ldots, nFrames - offd \tag{1}$$

The 2D probability histogram for the given offset distance is computed by dividing each element of the final 2D co-occurrence matrix by the sum of all the elements in the matrix.

$$P_{offd}(m, n) = \frac{C_{offd}(m, n)}{\sum_{c \in C_{offd}} C_{offd}} \tag{2}$$

The 2D probability matrix computed in (2) is converted to a 1D feature vector for learning purpose. In case of multi-offset distances, we have multiple 1D feature vectors, and different learning schemes are explored for learning these local patterns. Co-occurrence matrix-based statistics are the conventional features in several researches [7, 8, 14, 15]; however, these are not derived in our case. Instead, we use the co-occurrence matrix probabilities directly as the features since we wish to preserve the pair-wise intensity information in the order from 0-0 to 256-256. Statistical tools such as averaging would to some extent destroy the dynamic information in dynamic texture videos, as observed in the case of the averaged 3D-difference theoretic texture features [4] which retained information about the salient object in motion rather than the actual type of motion in the video. The advantages of using the probability values from the temporal co-occurrence matrix directly as features for classification can be summarized as:

1. Computationally too simple to calculate.
2. The intensity-to-intensity relationships over time are recorded as it is, so intensity-wise static and dynamic relationships are recorded starting from 0-0 to 256-256 pairings of intensities along the time axis.
3. The large feature dimension problem is compensated by the fast execution of the program.

4. Use of multiple offset distances allows for flexibility in the speed of motion of the dynamic textures which varies for different videos.

3 Proposed Methodology for Dynamic Texture Recognition by Local Pattern Matching

Figure 1 shows the overall feature extraction and classification process for our method. The frames are converted to grayscale, and for twenty different offset distances $offd = 1, 2, \ldots, 20$, the 256×256 temporal co-occurrence matrices are computed that are subsequently converted from 2D to 1D. The ensemble of decision trees classifier is used to learn and classify the extracted features.

The computation of the temporal co-occurrence probability matrix is described in Fig. 2. It is achieved by considering the combination of an intensity of the current frame and the intensity of a subsequent frame that is *offd* distance apart along the time axis, and by incrementing the cell assigned to this combination in the co-occurrence matrix by one. This is visually demonstrated in Fig. 3 where for an offset distance of 1, the combination (1, 2) occurs in consequent video frames (in top row) and corresponding (1, 2) cell in the co-occurrence matrix (bottom row) is incremented by 1.

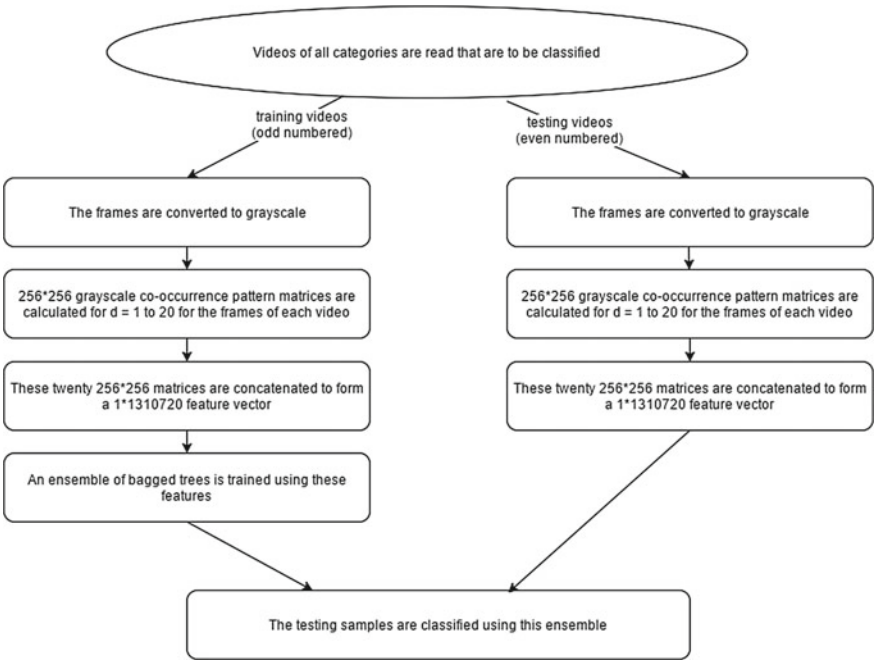


Fig. 1 Flowchart for the feature extraction and classification for the proposed method

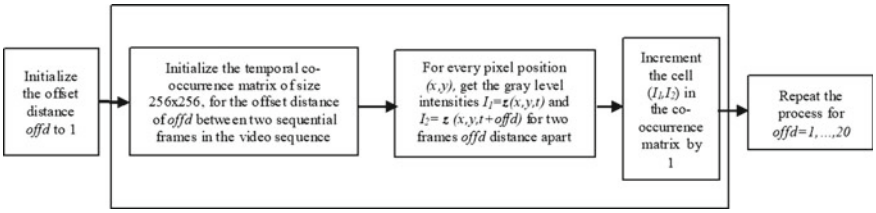


Fig. 2 Temporal co-occurrence probability matrix computation

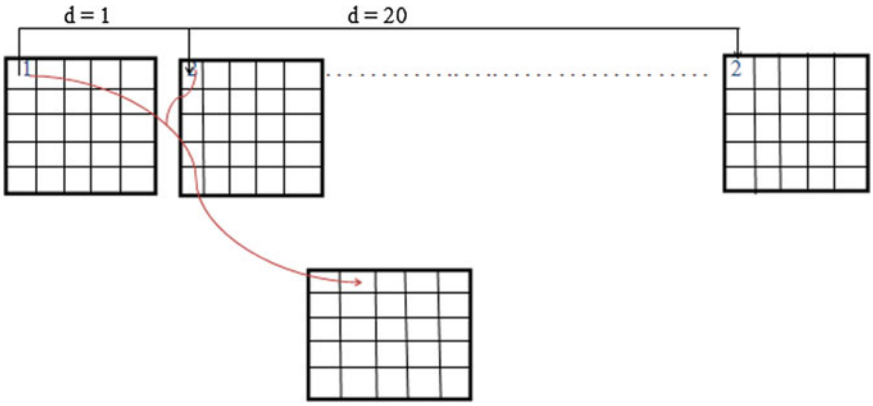


Fig. 3 An illustration of the updation of the cell (1, 2) in the temporal co-occurrence matrix in the bottom row, due to an occurrence of the combination of intensities (1, 2) in the frame sequence in the top row, when the two frames are separated by an offset distance equal to 1

Figure 4 shows the twenty co-occurrence matrices computed for two challenging video sequences belonging to the same class '*Elevator.*' Snapshots of the original video sequence are also shown. The two videos in Fig. 4a, b, as observed from the sequence of frames look visually very different and also show a difference in the speed of motion, with the *Elevator'* in the second video in Fig. 4b being much faster.

Due to differences in the rate of motion, an exact match of the twenty co-occurrence matrices is not possible, as observed from Fig. 4c, d. However, there could be local matches between the co-occurrence matrices of the training and the test videos. To quantify, measure, and detect this match at a local level, a distance grid of 20×20 cells is formed where the city block distance between every possible combination of the twenty co-occurrence matrices (in the form of ID feature vectors) of the training and testing samples is computed as shown in Fig. 4e. A similar classification process in [21] was found to improve results drastically when all the SIFT keypoints in one facial image were matched with all the SIFT keypoints in the second facial image.

The color visuals of the two '*Elevator*' videos are shown in Fig. 4f in order to appreciate the visual difference between the two videos both in the scene layout and the colors involved. The net distances from all such 20×20 grids are summed up, the

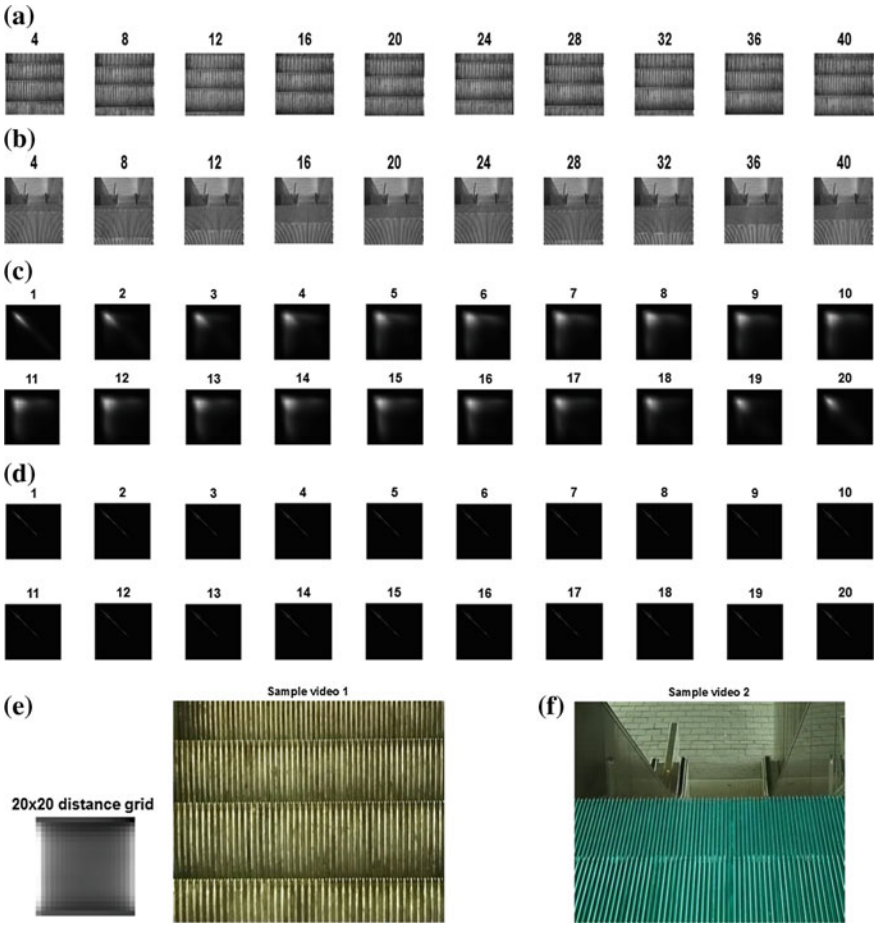


Fig. 4 Visual depiction of the twenty co-occurrence matrices of two different sample videos belonging to the same class '*Elevator*': **a** sample video 1 with its frame numbers, **b** sample video 2 with its frame numbers, **c** twenty co-occurrence matrices of sample video 1 shown numbered from 1 to 20, **d** twenty co-occurrence matrices of sample video 2 shown numbered from 1 to 20, **e** 20×20 distance grid showing the matches between local patterns of co-occurrence matrices, **f** two frames of the actual videos shown in color, to appreciate the visual difference

minimum summation distance with respect to all training samples decides the closest training sample, and its class is assigned to the test sample under consideration. The whole matching process is summed up in the following equation.

$$Class^{(Test)} = \underset{\forall Training}{\arg\min} \sum_{offd1} \sum_{offd2} \left| P_{offd1}^{(Training)} - P_{offd2}^{(Test)} \right| \tag{3}$$

where $offd1 = 1, 2, \ldots, 20$; $offd2 = 1, 2, \ldots, 20$.

The sum of city block distances for matching grid-based patterns has been effectively used before for facial expression analysis [22], though distance grids for matching individual local patterns were not constructed.

The overall methodology for the proposed dynamic texture recognition using co-occurrence matrices is summarized below.

1. Consider a video sequence having *Nframes* number of frames.
2. Convert the color space into grayscale values.
3. Initialize twenty 256×256 temporal co-occurrence matrices (for twenty offset distances $offd = 1, 2, \ldots, 20$) to all zeros.
4. For the offset distance *offd*, for every frame pair $(t, t+offd)$ in the video sequence, for $t = 1, \ldots, Nframes\text{-}offd$, update the *offd*th temporal co-occurrence matrix by counting the number of pair-wise occurrences of intensities at every pixel position in the tth and the $(t+offd)$th frames. The corresponding cell in the co-occurrence matrix is updated by the count as illustrated in Fig. 3.
5. Repeat step 4 for the twenty offset distances $offd = 1, 2, \ldots, 20$.
6. Convert each matrix to a complete probability distribution by dividing each element in the matrix by the sum of all elements in the matrix.
7. Reshape each of the twenty 2D probability matrices into 1D vector.
8. Repeat for all training samples.
9. The twenty 1D feature vectors are given to a 20×20 distance classifier that matches every combination pattern between the training template and the test vector.
10. Summation of the distance from all the 20×20 nodes is followed by comparison among all the training templates for a best match denoted by the minimum of the summation distance.
11. Alternatively, the entire feature space is converted to a 1D feature vector for learning the ensemble of bagged decision trees classifier.

4 Experimental Results and Discussions

The Dyntex++ dataset [23] contains a collection of 36 dynamic texture classes of a varied sort such as waterfall, boiling water, river water, fountain, underwater life, foliage, smoke. After eliminating those classes that contain only a single video sequence, we have 32 classes of dynamic textures with 341 video sequences in all, for our experiments. Some snapshots of dynamic textures are shown in Fig. 5.

Alternate video sequences are used for training and testing, and both validation and cross-validation are carried out for assessing the performance of the proposed dynamic texture recognition. The software implementation is done in MATLAB 7.10 version on a 2.7 GHz i5 processor. The proposed method is implemented as per the procedure outlined in the previous section. The results obtained are compiled in Table 1. As observed, the highest classification accuracies are recorded for the proposed temporal co-occurrence features using the multi-offset classification

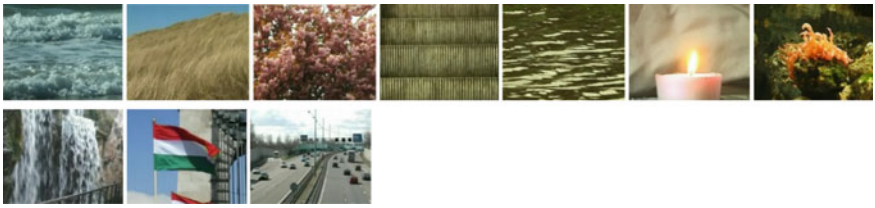


Fig. 5 Some samples of dynamic texture videos from the Dyntex++ dataset [23]

Table 1 Performance comparison for dynamic texture recognition for the Dyntex++ dataset: execution time (in secs) and percentage accuracy (%) for both validation and cross-validation (*best performance highlighted in bold*)

Method	Execution time in secs (*on an average for a video having 250 frames of size 288 × 352*)	Feature dimension	Classifier	Validation accuracy (%)	Cross-validation accuracy (%)
Proposed temporal co-occurrence probability features (*concatenated*)	41.51 s	1310270	An ensemble of bagged trees (50 trees)	64.38	64.9
Proposed temporal co-occurrence probability features (*multi-offset*)	**41.51 s**	20 × 65536	Distance summation of 20 × 20 city block distance classifiers	**68.75**	**66.43**
Deep learning by multi-dimensional CNN [10]	>1 h for each video cropped to 50 × 50 × 50	–	Multi-dimensional CNN	3.125	3.125
LBP-TOP [2]	621.41 s	177	Ensemble of bagged trees (50 trees)	68.12	62.9
Chetverikov and Péteri [6]	1680 s	20 × 20	Martin distance	20	18
Temporal co-occurrence statistics: entropy [13]	43 s	20	Ensemble of bagged trees (50 trees)	20.73	18.03

scheme. The best results are observed for the distance grid classifier, which implies that matching local co-occurrence patterns improves results.

We compare our results with some of the existing methods for dynamic texture recognition in the literature: Deep learning with multi-dimensional (3D) CNN [10],

LBP-TOP spatiotemporal features [2], Chetverikov and Péteri's stochastic dynamical systems [6], and the temporal co-occurrence-based statistic—entropy measure [13]. The LBP-TOP features give the second-best results. The ensemble of bagged decision trees used for learning and classification in Table 1 has 50 trees. Our method takes only approximately 41 s for execution, the least execution time among all methods. The CNN-based deep learning takes more than one hour to execute for each video cropped to $50 \times 50 \times 50$, and the absence of sufficient training is the reason for its low accuracies since we use the raw video samples without any cropping or dissection. The high accuracies obtained by our approach inspire us to work further in the area of temporal co-occurrence matrices, with a focus on feature reduction and the use of nonlinear classifiers for local pattern matching. This forms the future scope of our work.

5 Conclusions

The temporal intensity co-occurrence matrix probabilities are used as the features in this paper, for dynamic texture recognition. Twenty offset distances are used for the computation that yields twenty 2D probability matrices, each of which is converted to a 1D vector. The twenty 1D feature vectors form the feature space. A 20×20 grid of distance classifiers measures the closeness of the local patterns for multi-offset co-occurrence probability vectors. The simplicity of the computations compensates for the large feature dimension, and the highest recognition rate is achieved when compared to the state-of-the-art for our experiments on the benchmark Dyntex++ dataset.

References

1. Qi, X., Li, C.G., Zhao, G., Hong, X., Pietikainen, M.: Dynamic texture and scene classification by transferring deep image features. Neurocomputing **171**, 1230–1241 (2016)
2. Ji, S., Wei, X., Yang, M., Kai, Yu.: 3D convolutional neural network for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
3. Guoying, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. Ieee Trans. Pattern Anal. Mach. Intell. **29**(6) (2007)
4. Susan, S., Chakre, R.: 3D-difference theoretic texture features for dynamic face recognition. In: 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), pp. 227–232. IEEE (2016)
5. Hajati, F., Tavakolian, M., Gheisari, S., Gao, Y., Mian, A.S.: Dynamic texture comparison using derivative sparse representation: application to video-based face recognition. IEEE Trans. Hum. Mach. Syst. (2017)
6. Chetverikov, D., Péteri, R.: A brief survey of dynamic texture description and recognition. Comput. Recogn. Syst. 17–26 (2005)
7. Péteri, R., Chetverikov, D.: Dynamic texture recognition using normal flow and texture regularity. In: Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2005), Estoril, Portugal, pp. 223–230 (2005)

8. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Dynamic scene recognition with complementary spatiotemporal features. IEEE Trans. Pattern Anal. Mach. Intell. **38**(12), 2389–2401 (2016)
9. Rivera, A., Chae, O.: Spatiotemporal directional number transitional graph for dynamic texture recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(10), 2146–2152 (2015)
10. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 2, pp. 11–11. IEEE (2001)
11. Fablet, R., Bouthemy, P.: Motion recognition using nonparametric image motion models estimated from temporal and multiscale co-occurrence statistics. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1619–1624 (2003)
12. Yang, F., Xia, G.S., Liu, G., Zhang, L., Huang, X.: Dynamic texture recognition by aggregating spatial and temporal features via ensemble SVMs. Neurocomputing **173**, 1310–1321 (2016)
13. Haralick, R.M.: Statistical and structural approaches to texture. Proc. IEEE **67**(5), 786–804 (1979)
14. Haralick, R.M., Shanmugam, K.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **6**, 610–621 (1973)
15. Susan, S., Hanmandlu, M.: A non-extensive entropy feature and its application to texture classification. Neurocomputing **120**, 214–225 (2013)
16. Susan, S., Hanmandlu, M.: Color texture recognition by color information fusion using the non-extensive entropy. Multidimension. Syst. Signal Proc. 1–16 (2017)
17. Susan, S., Hanmandlu, M.: Unsupervised detection of nonlinearity in motion using weighted average of non-extensive entropies. SIViP **9**(3), 511–525 (2015)
18. Nelson, R.C., Polana, R.: Qualitative recognition of motion using temporal texture. CVGIP: Image Underst. **56**(1), 78–89 (1992)
19. Zalevsky, Z., Rivlin, E., Rudzsky, M.: Motion characterization from co-occurrence vector descriptor. Pattern Recogn. Lett. **26**(5), 533–543 (2005)
20. Bouthemy, P., Fablet, R.: Motion characterization from temporal cooccurrences of local motion-based measures for video indexing. In: Proceedings of Fourteenth International Conference on Pattern Recognition, vol. 1, pp. 905–908. IEEE (1998)
21. Susan, S., Jain, A., Sharma, A., Verma, S., Jain, S.: Fuzzy match index for scale-invariant feature transform (SIFT) features with application to face recognition with weak supervision. IET Image Proc. **9**(11), 951–958 (2015)
22. Susan, S., Kakkar, G.: Decoding facial expressions using a new normalized similarity index. In: 2015 Annual IEEE India Conference (INDICON), pp. 1–6. IEEE (2015)
23. Péteri, R., Fazekas, S., Huiskes, M.J.: DynTex: a comprehensive database of dynamic textures. Pattern Recogn. Lett. **31**(12), 1627–1632 (2010)

Multimodal Biometric Authentication System Using Hand Shape, Palm Print, and Hand Geometry

Gaurav Jaswal, Amit Kaul and Ravinder Nath

Abstract Developing a multimodal biometric system based on single-shot imaging (SSI) has recently grown interested in researchers worldwide. A palm region basically enriches with most discriminative features like lines, shape, and geometry which can be easily clubbed and captured together. In this work, feature-level fusion of hand shape, geometry, and palm print features has been performed. The extracted palm ROI samples undergo certain rotation and illumination effects that limit the matching performance. ROI samples are first geometrically aligned and then transformed into illumination-invariant form using CS-LBP. Further, local key points of transformed ROI images are extracted using SURF descriptor. In addition to this, a set of novel geometrical and shape features have also been computed from the hand registered image. All the three set of features are concatenated, and then, the highly uncorrelated features are selected from the fused feature set using sub-pattern PCA for classification. The performance of the proposed multimodal system is found to be superior to each of individual modality as well as reported state-of-the-art systems.

Keywords SURF · CS-LBP · PCA · Dual inverse

G. Jaswal (✉) · A. Kaul · R. Nath
Signal Processing and Biomedical Instrumentation Laboratory, Electrical Engineering Department, National Institute of Technology, Hamirpur, Himachal Pradesh, India
e-mail: gauravjaswal@nith.ac.in; er.shgaurav@gmail.com

A. Kaul
e-mail: Kaul@nith.ac.in; amitkaul9@gmail.com

R. Nath
e-mail: nath@nith.ac.in; r.nath1964@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_42

1 Introduction

Nowadays, hand biometric-based systems are getting popular as they provide accurate recognition results even on low-resolution images, require economical touchless imaging sensors, and need lesser user cooperation [1]. The multiple hand patterns such as hand geometry, palm print, hand vein, finger dorsal knuckle print, fingerprint have supposedly unique anatomical structures for biometric applications. Out of these, the quality of palm surface is literally found well than a fingerprint and also consists of a bigger ROI with stable lines and creases [2]. Palm print images can be captured in a contactless manner using cheap low-resolution cameras and non-intrusive in nature. Also, geometrical and shape features of hand/finger (adults) do not change much with time [3]. The vein traits in hand are distinctive and difficult to spoof, while finger knuckle lies outer side of fingers and is naturally preserved [4]. However, the unimodal biometric system is not fully perfect to be used alone in criminal, immigration, defense type of security applications. The performance of single modality systems is always influenced by certain common parameters such as low-quality images, background situations, camera flash, spoof attacks. On the other hand, multimodal biometric solutions combine information extracted from several biometric traits, thereby exhibiting higher resilience against spoofing [5]. In order to achieve better cost-effectiveness in multimodal biometric recognition systems, it is desirable to reduce the number of sensors employed for image acquisition. This can be feasible if multiple biometric characteristics are part of single image. Thus, in order to augment user convenience as well as to extract multiple traits by single shot imaging, we propose to capture the whole frontal hand image in such a way that multiple traits like hand shape, palm print and hand geometry can be extracted. Figure 1 show the anatomical view of palm print and hand geometry traits. In particular, we have proposed to extract the palm print and shape and geometry from the hand image.

1.1 Contribution

The key contribution of this paper is summarized into following steps: In case of palm print, the contrast and sharpness of extracted ROI images are improved using contrast-limited adaptive histogram equalization (CLAHE). The enhanced ROIs are first geometrically aligned by the dual inverse algorithm and further encoded by CS-LBP into illumination-invariant representation. A feature descriptor known as SURF is used to detect stable key points over transformed images. Moreover, efforts are being made to compute the geometric and shape features over hand contour. All the extracted features are first normalized and then concatenated using feature-level fusion. A compact representation of the extended feature space is given by SpPCA that gives uncorrelated feature coefficients for feature classification. The complete architecture of the proposed biometric system is given in Fig. 2.

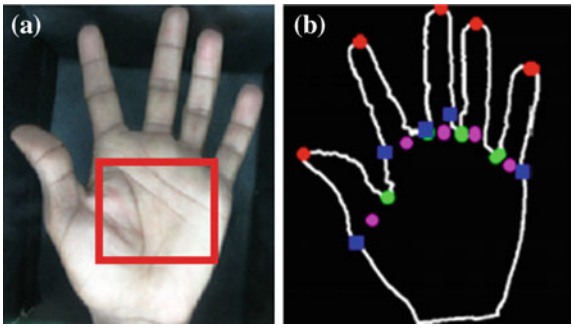


Fig. 1 Biometric traits anatomy: **a** palm print and **b** hand geometry

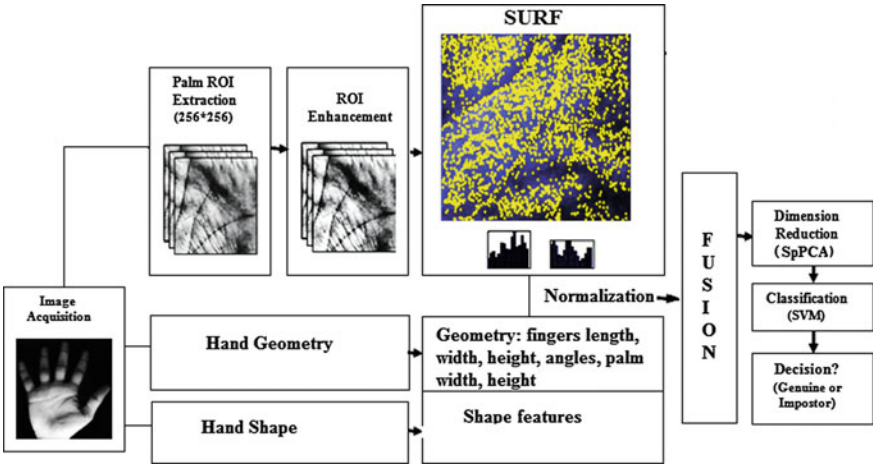


Fig. 2 Proposed architecture of multimodal biometric system

The publicly available GPDS-CL1 palm print and Bosphorus hand databases are used for performance evaluation in terms of EER, DI, and CRR. The rest of the paper is organized into totally five sections: Sect. 2 describes the state-of-the-art work related to palm print and hand shape/geometry. The proposed multimodal system is presented in Sect. 3. Further, the experimental results are discussed in Sect. 4. Finally, the important findings are summarized in the very last section.

2 Literature Survey

The random lines, geometry, shape, and texture over palm region of hand have a lot of discrimination ability to be broadly recognized as a biometric identifier. Initially, the inked palm print images were used for offline recognition systems. Similarly, contact-

based flatbed scanners were suggested by various researchers. Currently, most of the studies are focussing on contactless/peg-free imaging devices, which improve user-friendliness and enhance the wider span of applications. In [5], Sobel edge detector of multiple kernels and orientations was applied to extract palm edge features. In [6], authors made efforts to convert palm print image from a spatial domain to a frequency domain by using discrete Fourier transform (DFT). In addition to this, various coding methods like PalmCode [2], OrdinalCode [7], CompCode [3] are well reported in the literature. In [8], a total of 34 geometric features were computed from the palm region of the hand. In [9], authors focused to extract geometric features and vein patterns and matched using principal component analysis. Besides the unimodal recognition approaches, fusion-based multimodal approaches have also been explored such as palm print and finger geometry (matching score) [10], palm print and hand geometry (feature level) [11], hand geometry and shape (score level) [12].

3 Proposed Multimodal Biometric System

The main objective of the proposed approach is to present a single-shot simultaneous acquisition-based mechanism to improve the cost and user cooperation of multimodal biometric systems. This is possible by integrating hand geometry, palm print, and hand shape features because they lie close to each other. The following are the main stages in our proposed multimodal biometric system.

3.1 Preprocessing Methods for Palm Print Recognition

The low-resolution images are easily corrupted due to sensor-level noise. Secondly, poor segmentation/alignment and large-size template of original image increase the computation cost. Thus, it becomes important to overcome these shortcomings at initial level. In this section, the main preprocessing approaches are involved:

1. **ROI Extraction**: Most of palm print ROI extraction algorithms share a common viewpoint to set the position of the segmented square area. In [2], authors measured two valley points near fingers' fissure and a fixed size square is cropped. Likewise in [3], authors obtained two midpoints at the end surface of middle and little fingers. In this work, palm ROI is extracted using the algorithm proposed in [13]. The original dimension of the RGB image (GPDS-CL1 database) is 1600×1200. A ROI of square size (412×412) has been cropped from the original image as shown in Fig. 3. Then, images are enhanced using CLAHE as mentioned in [13].
2. **Image Transformation 1**: Inverse compositional algorithm has been employed in various computer vision tasks for image registration so that any two images match as best possible. In this work, dual inverse compositional (DIC) method

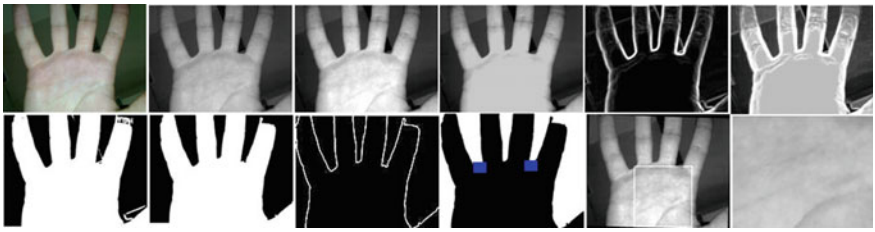


Fig. 3 Palm print ROI extraction (sample image taken from GPDS-CL1)

based on geometric and photometry-based image alignment has been used [14]. This method aims to align two images by computing both a warp and a parametric photometric transformation that depends upon the intensity of pixels. Let us consider two palm print images, namely target (I_T) and source (I_S). Now, our objective is to align a target image (I_T) to a source image (I_S) by computing the two main parameters: g of geometric registration $\hat{G}$ (:$\hat{g}$) and p of photometric registration $\hat{P}$(,:p). The other possible inputs to this algorithm are: ROI in source image (I_S), initial values of g and p. Now, estimate the transformation parameters p and g that minimize the photometry error between I_S and I_T.

$$\sum\nolimits_{q \in R} \left\| I_{s(q)} - p(I_T(G(q;\, g));\, P) \right\| \tag{1}$$

Next, compute the incremental wrap and photometry parameters and solve the joint incremental parameters Ψ_{gp}. Now, update the parameter p and g until:

$$g \leftarrow U_g\left(g,\, \Psi_g\right) \tag{2}$$

$$p \leftarrow U_p\left(p,\, \Psi_p\right) \tag{3}$$

The example aligned images by computing two parameters (g, p) are shown in Fig. 4.

3. **Image Transformation 2**: In order to extract dense palm features under different resolution, we adopted a computationally efficient descriptor known as center symmetric local binary pattern (CS-LBP) [15]. It encapsulates the gradient information present in the images by differentiating the pixels intensities in a local image neighborhood. The CS-LBP based image encoding is derived from the local binary pattern (LBP) features by assigning a binary code to each pixel of the image. In particular, all the neighborhood pixel intensities are thresholded with the gray scale level of center pixel. This joint representation of such N binary values results into a decimal code. In this work, a neighborhood of 8 pixels (N = 8) and radius (R = 2) centered at the current pixel are considered. In particular, the difference between pixel intensities of *i*th and (i + N/2)th pixels, denoted by g_i and $g_i + (N/2)$, respectively, is considered and a binary code is assigned to each pixel in the enhanced ROI $I_{(x;\, y)}$ [14]. Finally, this generates the CS-LBP image

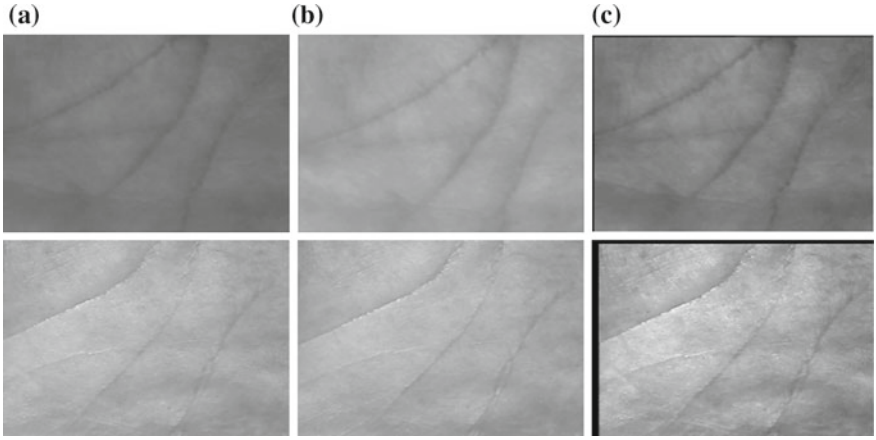


Fig. 4 DIC based geometric transformation: **a** I_S, **b** I_T, and **c** transformed image (sample GPDS-CL1 image)

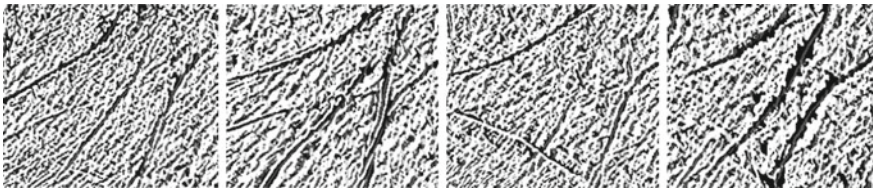


Fig. 5 CS-LBP-based image transformation: sample GPDS-CL1 images

which is an encoded template of original palm with values vary from 0 to 16 and is further used for SURF matching (Fig. 5).

3.2 Preprocessing Methods for Hand Shape and Geometry Recognition

The procedure of preprocessing scheme for hand image involves: background removal, binarization, hand orientation registration, contour detection with reference point, peaks, and valleys detection, and landmark feature points determination. To do this, the original color image (Bosphorus hand database) is converted into gray scale and then 2D Gaussian filter with size (3×3) is applied to overcome the input noise and other irregularities. Next, the grayscale image is converted into black-and-white image based on the knowledge of foreground and background pixels. Further, the hand is rotated in such a way that the main axis of middle finger becomes longitudinal, as shown in Fig. 6.

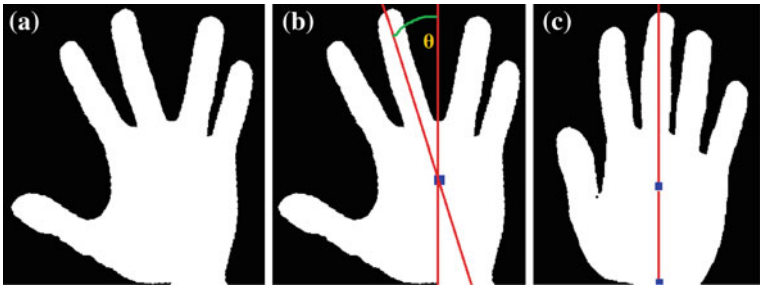


Fig. 6 Hand registration steps

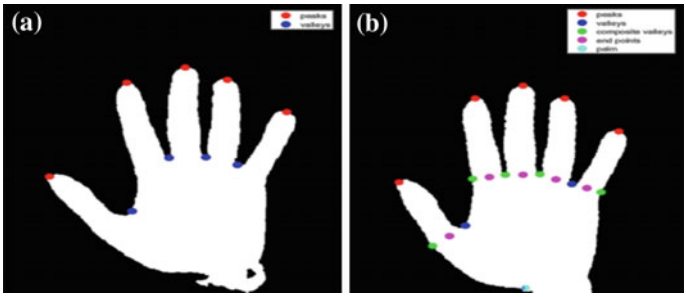


Fig. 7 Peaks and valley point detection

In the next step, we have found out a stable reference point near the wrist of hand to attain the invariance against translation and to detect the valleys and peaks of the fingers. To efficiently calculate the all possible five peaks and four valleys between the fingers, a distance map is generated. The minima and maxima of the distance map are defined as valleys and peaks of the finger. The resultant peaks and valleys are shown in Fig. 7.

3.3 Feature Extraction

To analysis countless lines, points, and geometric features in a hand image, the following methods have been used:

4. **Palm Print Feature Extraction by Speeded-up Robust Features (SURF)**: In the past few years, SURF has been used for various machine learning and object recognition problems [16]. It extracts scale-, rotation-, and translation-invariant features of the images which are very distinctive and stable key points for matching. It is basically an approximation of SIFT, but performs faster than SIFT without reducing the quality of the detected points. The four main steps for computing the SURF descriptor for image are: key point detector, key point

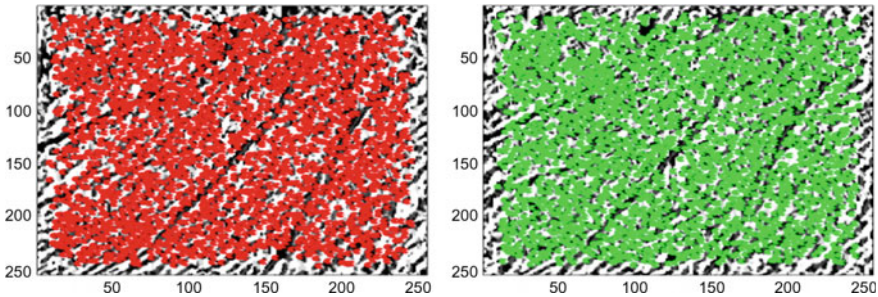


Fig. 8 Detected SURF key points of corresponding palm prints

descriptor, orientation assignment, and descriptor. In brief, SURF provides an imprecise idea about difference of Gaussian with box filters. As an alternate of Gaussian averaging, the convolution of image with square is much speedy, even for different scales. A binary large object detector based on the Hessian matrix is selected to find the interest points in that region. In the next step, it computes wavelet responses in both transverse and longitudinal directions for orientation assignment. A region surrounding the key point is chosen and separated into subregions from which wavelet responses are computed and represented to acquire SURF feature descriptor. Finally, each of the descriptors from the subregions is summed up and thus a large descriptor of dimension 64 is obtained, named as SURF descriptor. In this work, SURF is used to extract palm print key points for one-to-one matching (Fig. 8).

5. **Hand Contour Shape Extraction by HOG**: The contour of hand is basically the outer boundary which defines a continuous relation among the pixels over it. But the shape of hand changes abruptly with the movement of fingers and affects the recognition. To control such problems, a deformable shape descriptor has been used namely nr-HOG (non-rigid histogram of gradients) which can detect the shape regions using non-rigid grid criteria [17]. In this method, the contour image of hand is first obtained as mentioned in Sect. 3.2 and then HOG descriptor is applied on contour image only. The geometric centroid of a region is calculated using number of shape pixels, and the regions centered over geometric centroid are described using non-rigid HOG. In this work, an image partitioning algorithm [17] described earlier has been used. The three image decomposition levels are selected in which each overlapping region (block) is split into corresponding four subregions (cells) and then location of geometric centroid is assumed. Finally, the regions that are centered over geometric centroid are selected and 3 * 3 cell grid is used to compute HOG (9-bin histogram) features for each cell. The cell grid basically depends upon image size and overlapping parameter. An example of non-rigid HOG-based feature extraction (3 * 3 * 9 = 81 features) is shown in Fig. 9 in which the pixel gradient information (magnitude and orientation) is computed over the contour of hand.

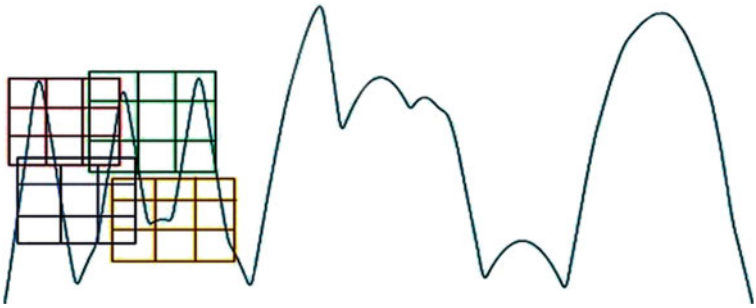


Fig. 9 HOG (3 × 3) grid on radial distance obtained by hand contour

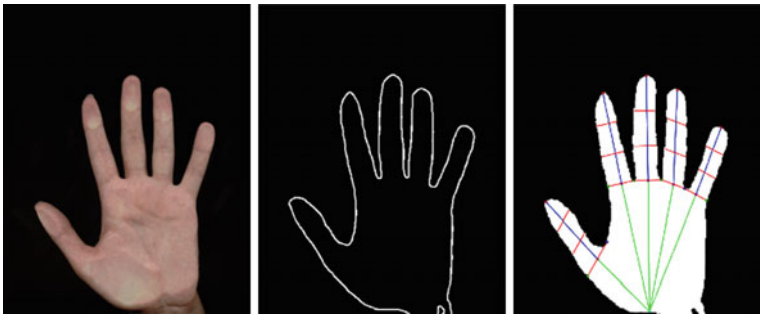


Fig. 10 Hand geometry measurements

6. **Hand Geometry Extraction**: The finger/palm length, width, thickness, area, curvature, angle, etc., usually involve in determining the geometry of hand. These physical dimensions are measured over several points as one can see the feature point detection in Fig. 7. In this work, we have computed 45 set of features including: finger lengths, finger widths, palm lengths, angles, variance of finger lengths, variance of finger widths, variance of palm lengths, ratio of consecutive finger widths, and ratio of consecutive palm lengths. In Fig. 10, we have tried to draw all these measurements over hand feature points.

3.4 Normalization and Feature Fusion

In feature-level fusion, the data obtained from different modalities or different feature vectors extracted from single modality can be concatenated using certain rules, etc. [1]. However, the feature vectors need to be transformed to a common range before concatenation. In the proposed algorithm, a min–max normalization method is applied to ensure the same numerical range and scales of feature sets. Let $X_{128} = \{x_1, x_2, \ldots, x_n\}$, $Y_{81} = \{y_1, y_2, \ldots, y_n\}$ and $Z_{45} = \{z_1, z_2, \ldots, z_n\}$ denote the fea-

ture sets (X, Y, Z € R^N) obtained from SURF, shape (nr-HOG), and hand geometry methods. Then, the feature vectors are normalized and concatenation of these feature vectors X, Y, and Z can be performed using simple sum rule, such as

$$Sum\ Rule : S_i = \left[\{X_i + Y_i + Z_i\}_{i=1}^{N}\right]_{k=1}^{S} \tag{4}$$

where S_i represents fused feature vector. But, the features extracted from multiple sources may have incompatible information and their fusion may lead to high dimensionality. High-dimensional features may consist of noisy and irrelevant data that decreases the performance of a classifier. Therefore, before applying a match between corresponding feature vectors, a compact representation of each fused feature vector is given by its projection onto the space expanded by the principal component analysis (PCA).

3.5 Sub-pattern-Based PCA (SpPCA) and SVM-Based Classification

To keep in view that local feature in image can perform better classification than traditional PCA approach, a robust method known as sub-pattern-based PCA has been employed [18]. In this method, PCA is directly applied on equally sized sub-patterns of image, and their local sub-features are autonomously detected from respective local projection sub-vectors. For recognition, it is important to compute weight vectors of the test sub-images using the same eigenvectors. The set of fused features of hand (palm, hand geometry, and shape) identified by the SpPCA are used to train a binary classifier (SVM), which (during the testing phase) determines whether the input image belongs to a real palm or a fake one.

4 Experimental Results and Discussions

The palm print database is obtained from GPDS-CL1 hand database for the validity of the proposed work. It contains 1000 hand images (right hand) collected from 100 users which are first cropped and further transformed using DIC and CS-LBP based methods. On the other side, hand shape and hand geometry recognition are performed on Bosphorus hand database. The Bosphorus database contains a total of 660 subjects in which 6 images per individual are collected. In distinguishing experiments, the individual and integrated performance of three modalities (palm print, hand shape, and hand geometry) has been evaluated in identification and verification mode. The performance parameters in this study are false acceptance rate (FAR), correct recognition rate (CRR), false reject rate (FRR), decidability index (DI), and equal error rate (ERR). The results are evaluated under equal number of training and

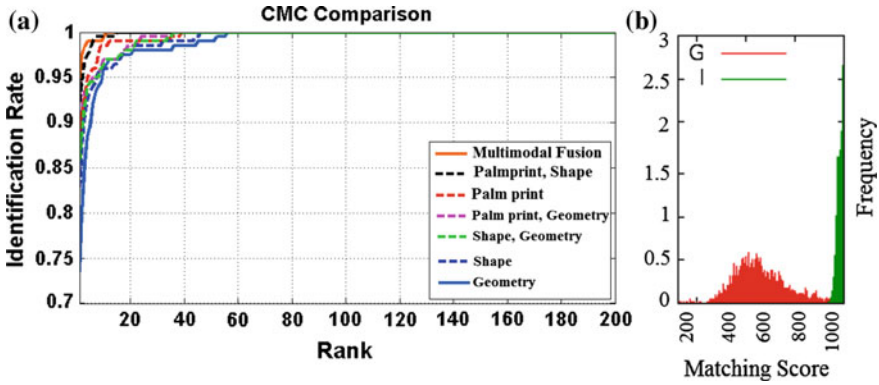


Fig. 11 **a** CMC-based performance analysis; **b** genuine versus impostor score

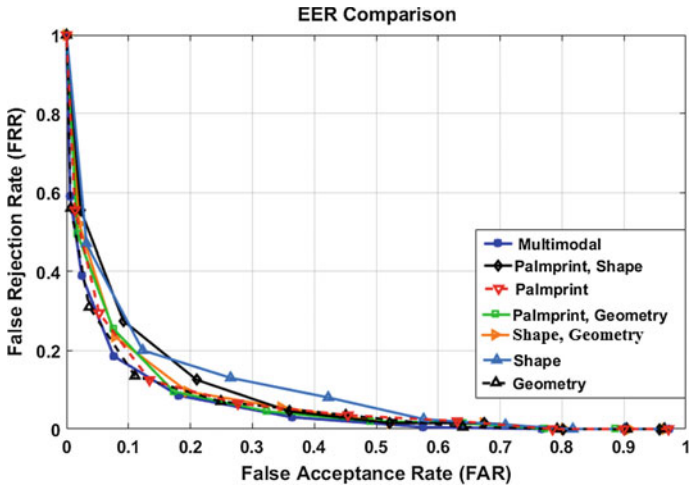


Fig. 12 EER-based performance analysis

test samples, and varying projection vectors. The CMC and EER curves for all the tests are shown in Figs. 11a and 12, while detailed description is presented in Table 1. We have also shown the genuine versus imposters score distribution graph in Fig. 11b. The proposed prototype system is executed in MATLAB language (R2017a), using a machine outfitted with i5 processor (2.33 GHz) with 4 GB Ram, and open-source Windows 10 as operating system. For better justification of the proposed method, the following experiments are conducted.

- Experiment-1: At first, GPDS-CL1 palm print database is considered and the performance of the proposed palm print algorithm is tested. The objective of this test is to compare the performance of the proposed method with and without applying the transformations. Out of ten images per subject, five training and five testing

Table 1 Comparative analysis between the proposed system and state-of-the-art methods

Dataset	Approaches	CRR (%)	ERR (%)	Speed (ms)
CASIA	PalmCode [3]	GAR-98	0.6	1100
800 images (self)	G2D2PCA [19]	99	–	5800
PolyU	SURF [16]	99.98	0.021 (FAR); 0.01 (FRR)	–
1500 images (self)	Gupta et al. [9]	90.66	11.93	–
GPDS-CL1 (Full)	Proposed—palm print system	99.26	1.022	1821
GPDS-CL1 (Full)	Palm print (without using DIC, CS-LBP)	93.44	4.22	1726
Bosphorus hand	Proposed—hand shape system	92.32	7.55	1165
Bosphorus hand	Proposed—hand geometry system	88.18	10.62	2050
Palm print, shape	Proposed—bimodal fusion	99.56	1.08	–
Palm print, geometry	Proposed—bimodal fusion	95.52	6.18	–
Shape, geometry	Proposed—bimodal fusion	90.83	10.20	–
Palm print, shape, geometry	Proposed—multimodal fusion	100	1.68	–

images are selected and a total of 2500 genuine and 25,000 impostor matchings are obtained. The important findings from the first test demonstrate that: (a) the proposed palm print schemes when applied in conjunction with transformation methods have shown significant improvement in overall results. It has achieved EER of 1.55%, CRR of 99.26%, and DI of 2.65. (b) The SURF-based descriptor extracts a huge number of stable key points over CS-LBP-based transformed ROI images.

- Experiment-2: Among hand shape and geometry methods, nr-HOG shape signifies the superior results over hand geometry performance. In case of performance of hand geometry methods, the distance measurements on hand get affected due to instable gap between fingers. The hand geometry system achieves CRR-88.18%; EER-10.62%; DI-1.98 with SVM classification over Bosphorus dataset. As far as the performance of HOG-based shape analysis, the results (CRR-92.32%; EER-7.55%; speed-1165 ms) have shown overall improvement with the same dataset. In this test, a total of 5940 genuine and 3,914,460 imposter matchings are obtained for both shape- and geometry-based analysis.

- Experiment-3: Multimodal score-level fusion is always advantageous for performance improvement because we can easily assign weights to each method. But irrespective of such benefits, it shows higher computational effects, specifically when feature sets of different dimensions are added. In order to overcome this situation, SpPCA-based dimension reduction methods have been incorporated, which not only achieves superior performance but provides time-efficient recognition results. In this test, we have performed different biometric fusions to justify the role of feature-level fusion for our proposed system. All the multimodal fusion tests are performed using virtual combination of GPDS-CL1 and Bosphorus datasets. We have evaluated the performance of bimodal systems, such as palm print and hand shape, hand shape and hand geometry, and palm print and hand geometry. Finally, we have evaluated the proposed multimodal biometric system which is the feature-level fusion of palm print, hand shape, and hand geometry traits. In this test, we have obtained 900 genuine and 89,100 imposter matchings by establishing the virtual combination of GPDS-CL1 and Bosphorus datasets. After doing rigorous experimentation, the following conclusions are drawn: Among the performed bimodal combinations, the fusion of palm print and hand shape achieves overall higher performance in terms of CRR-99.56%, EER-1.08%, and DI-3.02%. However, the proposed multimodal biometric systems surpass the other performed tests (CRR-100%, EER-1.68%, and DI-3.02) and justify its importance to improve the performance of hand shape- and geometry-based recognition systems. In addition to this, the proposed approach has been compared to four current state-of-the-art systems.

5 Conclusion

This work has been carried out with the motive of developing a hand recognition system which can be used for various security applications such as transfer of payments to laborers through various levels of financial inclusion projects, immigration, and border control at airport. In this work, a multimodal recognition system based on feature-level fusion of normalized features of palm print, hand shape, and hand geometry traits has been proposed. The results are compared with other state-of-the-art systems and shown comparable performance in terms of CRR (100%), EER (1.68%), and DI (3.02).

References

1. Jaswal, G., Kaul, A., Nath, R.: Knuckle print biometrics and fusion schemes–overview, challenges, and solutions. ACM Comput. Surv. **49**(2), 34 (2016)
2. Zhang, D., Kong, W.K., You, J., Wong, M.: On-line palm print identification. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 1041–1050 (2003)

3. Yang, W., Huang, X., Zhou, F., Liao, Q.: Comparative competitive coding for personal identification by using finger vein and finger dorsal texture fusion. Inf. Sci. **268**, 20–32 (2014)
4. Kong, A.W., Zhang, D.: Competitive coding scheme for palm print verification, pp. 520–523 (2004)
5. Wong, A.K., Chekima, A., Dargham, J.A., Sainarayanan, G.: Palmprint identification using Sobel operator. In: 10th IEEE International Conference on Control, Automation, Robotics and Vision, pp. 1338–1341 (2008)
6. Li, W., Zhang, D., Xu, Z.: Palmprint identification by fourier transform. Int. J. Pattern Recog. Artif. Intell. 417–43 (2002)
7. Sun, Z., Tan, T., Wang, Y., Li, S.: Oridnal palm print representation for personal identification. Comput. Vis. Pattern Recog. 79–284 (2005)
8. Guo, J.M., Hasia, C.H., Liu, Y.F., Yu, J.C., Chu, M.H., Le, T.N.: Contact-free hand geometry-based identification system. Expert Syst. Appl. **39**(14), 11728–11736 (2012)
9. Gupta, P., Srivastava, S., Gupta, P.: An accurate infrared hand geometry and vein pattern based authentication system. Knowl. Based Syst. **103**, 143–155 (2016)
10. Zhu, L-q, Zhang, S-y: Multimodal biometric identification system based on finger geometry, knuckle print and palm print. Pattern Recogn. Lett. **31**, 1641–1649 (2010)
11. Kanhangad, V., Kumar, A., Zhang, D.: Combining 2d and 3d hand geometry features for biometric verification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 39–44 (2009)
12. Dubey, S., Singh, S.K., Saxena, R., Singh, R.K.: Identity verification using shape and geometry of human hands. Expert Syst. Appl. 821–832 (2015)
13. Jaswal, G., Kaul, A., Nath, R.: Palmprint and finger knuckle based person authentication with random forest via kernel-2DPCA. In: International Conference on Pattern Recognition and Machine Intelligence. Springer, Cham (2017)
14. Bartoli, A.: Groupwise geometric and photometric direct image registration. IEEE Trans. Pattern Anal. Mach. Intell. **30**(12), 2098–2108 (2008)
15. Heikkila, M., Pietikainen, M., Schmid, C.: Description of interest regions with center-symmetric local binary patterns. IN: ICVGIP 58–69 (2006)
16. Srinivas, B., Gupta, P.: Palm print based verification system using SURF features. Contemp. Comput. 250–262 (2009)
17. Almazán, J., Fornes, A., Valveny, E.: Deformable hog-based shape descriptor. In: 12th International Conference on Document Analysis and Recognition, pp. 1022–1026 (2013)
18. Chen, S., Zhu, Y.: Sub pattern based principle component analysis. Pattern Recogn. **37**(5), 1081–1083 (2004)
19. Pan, X., Ruan, Q.-Q.: Palm print recognition using Gabor feature-based (2D)2PCA. Neuro Comput. **71**(13), 3032–3036 (2008)

Digital Image Restoration of Historical Devanagari Manuscripts

Nidhi Dubey

Abstract We live in a world where lots of manuscripts were written in earlier times. There are lots of historical manuscripts which are in deteriorated form. There is a need to restore these manuscripts in order to preserve our cultural heritage and ancient knowledge for future generation. Our focus is to restore the manuscripts that have been deteriorated because of age. These manuscripts got crushes, turned yellowish and got torn. There are some existing techniques for background restoration, and lot of work has already done in removing the background deterioration but very less work is done on foreground of manuscript for removing the deterioration. Our focus is to restore the foreground of manuscript by completing the incomplete characters in a torn places of manuscript through restoration techniques and for performing the background restoration, we have used in built Otsu binarization thresholding method. We are applying image processing methods for restoring text, foreground and background on deteriorated manuscript images.

Keywords Manuscripts · Deteriorated · Image enhancement · Image refinement Binarization methods

1 Introduction

A manuscript is defined as any document written by hand, being printed or reproduced in some other way [1]. The deterioration of ancient manuscripts happens because of the presence of iron gall inks that results mainly from two phenomena: hydrolysis and oxidation. The deterioration of manuscripts can result in following forms of manuscripts that are described below in Fig. 1 [1].

Manuscripts are one of the very important aspects of medieval society. The preservation of these manuscripts is very important because these manuscripts contain lot

N. Dubey (✉)
Department of Computer Engineering and Application,
GLA University, Mathura, Uttar Pradesh, India
e-mail: dubey.nidhi52@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_43

Fig. 1 **a** Yellowish manuscript. **b** Torn manuscript. **c** Ink bleeds in manuscripts. **d** Cracked manuscripts. **e** Faded manuscripts. **f** Foxing in manuscripts

of ancient knowledge which is useful for future generation in order to know more about our ancient culture. Manuscript plays a "vital" role. The historians often work with handwritten documents and manuscripts kept in archives and rare book libraries. These manuscripts consist of those data that we can use in science, technology and tradition in future.

The motivation of our work is to separate both background and foreground in order to restore manuscript in order to enhance background and completing characters in background.

The digital image restoration of historical manuscripts deals with restoring the documents (i.e. manuscripts), that have deteriorated, through digital image processing. The deterioration happened to a manuscript basically results in uneven and degraded background, the contrast of manuscripts that result in darkening of background of an image, due to which the contrast between the foreground and background reduces.

The paper is described as follows. Section 2 discusses the previous work done on restoration of deteriorated manuscripts. Section 3 gives a brief idea of the proposed framework in the paper and also explains the proposed method through an algorithm and proposed model; Sect. 4 presents step-by-step results; Sect. 5 gives you information about the conclusion and future work of paper.

2 Related Work

The work of image restoration has been previously done by many of the researchers in order to provide the improved result of the degraded old manuscripts. There is rarer work done on improving and restoring the foreground of manuscript image but so much of work is done on restoring and improving the background of manuscript image. There are some methods/techniques that have been used by researchers in order to restore manuscript. Some of the most popular methods that worked on this problem are Otsu's thresholding technique [2], techniques that deals with entropy proposed by Kapur et al. [3], and the method deals with minimal error technique given by Kittler and Illingworth [4]. These entropy-based methods deal with noise inherent in the paper which were especially designed for segmentation of historical document. The direct image matching and directional wavelet methods were used by author Wang et al. [5], in order to separate bleed through text and background noise from text.

Wafa et al. [6] proposed a method in order to separate foreground and background from a manuscript image, for that they have presented a system comprising of a method coupling together a system which includes two techniques, i.e. hybrid method of background light intensity normalization algorithm and k-means clustering with maximum likelihood (ML) estimation. They have used background normalization algorithm for separating foreground and background. The work is performed such a way that firstly they have applied background normalization algorithm that separates foreground and background, then they focused on improving luminance and suppressing the distorted contrast. In order to correct this distortion on images on images using gamma correction and for adjusting the contrast, the contrast adjusting method is used. Now the focus was on segmenting a new enhanced foreground image into foreground/background. The paper does not focus on texture segmentation is the only weakness of this paper. The texture segmentation mainly helps in classifying the document into text, background and graphic that can be useful for indexing and retrieval system.

Ntogas et al. [7], they proposed a binarization method that is based on pure thresholding and filtering in order to discriminate and categorize the text and background in a manuscript image with a help of using some image processing algorithms with it. The main focus of their work was on text image binarization that is been considered one of the most innovative and new method for acquisition of historical digital images. The plus points given by them are that they gave a method that is robust and simple. The binarization method, they have used on pre-filtered historical image and then the simulation is done on them in order to produce a result from it.

Ketki et al. [8], they focused on those historical manuscripts that were written in Arabic language which is having uneven background and low contrast in order to improve manuscript that got deteriorated and degraded due to age. The proposed method consists of five methods that followed in an order like: foreground extraction, contrast adjustment, foreground–background segmentation and reconstruction of document image with smoothening.

Brown et al. [9], they have proposed a method that uses a 3D geometry of a document in order to restore those manuscript images whose shapes are distorted. The system used by them is physically based mass-spring particle system in order to provide guidelines to the iterative un-warping process.

There are some other techniques also that have been proposed by authors in [10–12] that also restore manuscripts that have been deteriorated because of age.

The comparative analysis is given in Table 1.

On the basis of above comparative analysis, our proposed methodology works on these findings yellowish background, disturbed contrast, crushes, and incomplete character in an image and has done the restoration of RGB manuscript image in two steps: first step applies Otsu binarization technique [2] to restore its background, and second step calculates threshold value for each character and performing template matching to complete it. The novelty of the proposed technology is that it separates both foreground and background, restoring background along with that completing all incomplete characters in an image.

3 Proposed Work

3.1 Data Set

The data set has been taken from Vrindavan Research Institute, Mathura [13], which consists of some degraded old images of manuscripts, which is written in a Devanagari. Since the nineteenth century, it has been the most commonly used script for writing Sanskrit. This language was formerly used to write Gujarati, because it is the standardized script for writing the Hindi language. The data set is maintained in such a way that it will be containing images of all characters that are in Devanagari. There are some issues that came across while making a data set for this methodology. The Devanagari is handwritten, so every image needs its own data set that will be made from it. There are some images that are of similar type and having similar type of handwriting, so for that a common data set can be created to complete the incomplete characters in a manuscript image.

3.2 Algorithm

- Take a degraded image of an old Devanagari manuscript as an input.
- Firstly, we will work on foreground restoration of a manuscript image. For that, we will find out those places where the deterioration happened to an image. The pixel positions are identified in an image where the characters are partially incomplete in a manuscript.
- Now, we will work on each deteriorated position one by one.

Table 1 Comparative analysis

Comparative analysis						
S. no.	Title	Authors	Objective/research questions	Quantitative/quality review	Findings	Limitations
1	Restoration of degraded historical document image	B. Gangamma, Srikanta Murthy K, Arun Vikas Singh	Enhancing degraded noisy images	Spatial domain and set theory operations	Eliminates noise, uneven background and enhances contrast	Does not work for separation of foreground and background of manuscript
2	A binarization algorithm for historical manuscripts	Ntogas, Nikolaos Ventaz, Dimitos	Six categories were described on image based on text image binarization, image type classification, image preparation. according to image condition	Binarization procedure for pre-filtered historical manuscripts	Discriminating text from the background based on pure thresholding and filtering combined with image processing technique	Does not work for all types of images The results have shown improved image quality for the for the six categories of document images which were described by their separate characteristics
3	An adaptive binarization technique for low-quality historical documents	Basilios Gatos, Ioannis Pratikakis, Stavros J. Perantonis	In historical documents preservations on meaningful information on text of low quality	Preprocessing procedure uses low-pass Wiener filter, Niblack's approach and post-processing step to improve the quality of text regions and preserve stroke connectivity	Estimation of foreground regions, background surface calculation	Text areas are located if the distance of the original image with the calculated background is over a threshold
4	Enhancement of writings on a damaged medieval manuscript using ultraviolet imaging	Isabelle Montania, Eric Sapina, Alexandre Pahudb, Pierre Margota	Worn and illegible portions on text were enhanced of text The transcription of the text became difficult because of deterioration due to ink and parchment	White light revealed different types of damages, either linked to the ink or to the parchment	More the paper is degraded, more it becomes UV absorbent	The area that were more focused was where the ink was fainted on non-degraded support. The presence of contrast was always there but spatial and tonal irregularities were with weaker intensity
5	A methodology for the separation of fore-ground/background in Arabic historical manuscripts using hybrid methods	Wafa Boussellaa Abderrazak Zahour Adel Alimi	A segmentation on foreground–back-ground of colour historical Arabic manuscripts	Post-processing of obtained foreground with double contrast adjustment. Foreground extraction with iterative light intensity normalization algorithm	Foreground/ background segmentation suited for Arabic documents manuscripts distorted with show-through effects and uneven background	An improvement on texture segmentation method is done

- We will calculate the threshold value for each character by calculating Euclidian distance for each character so that we can set some threshold value for replacing an incomplete character by that same complete character that is stored in a dataset in an image.
- The template matching will be performed in order to pass characters images from data set as a template, so that we get to know about the character which is missing.
- Now, we will apply superimposing concept here. We got to know the character and we got to know the position also where we want to complete the character. So, we will superimpose the incomplete character with complete character in manuscript image.
- Above steps can be repeatedly done for all positions in a manuscript image where the deterioration has happened.
- We will get a manuscript image that is free from foreground deterioration. The foreground of an initial image of manuscript is now having all complete characters.
- The next step is to restore the background of an image. For improving the background and yellowish colour of an image, we will perform binarization using an Otsu binarization technique on an image.
- Finally, we will get an image that is free from background deterioration.

3.3 Technique

The proposed approach mainly focuses on foreground restoration on a manuscript image written in Devanagari. There are so many techniques that have been applied to old deteriorated manuscript images, in order to restore its background. Firstly, we will discuss the techniques and concept through which we can restore the foreground of an image, by completing the incomplete characters in a manuscript image. Then, we will move towards the restoration of background of an image that will help in removing the yellowish background, blurring on an old degraded manuscript and crushes on manuscripts. The main aim of our proposed work is to restore the characters of manuscripts which will help in improving the clarity of content in Devanagari manuscripts.

3.3.1 Foreground Restoration for Character Completion

In this category, our focus will be on restoring the foreground of manuscript image by completing the complete character in an image

Finding the position of deterioration
The deterioration on manuscript image will be removed in such a way that firstly we need to by identify the place of deterioration in a manuscript image. For identifying the places of deterioration in an image, we have to firstly convert an RGB manuscript image into a binary image by to some threshold value so that only the deterioration

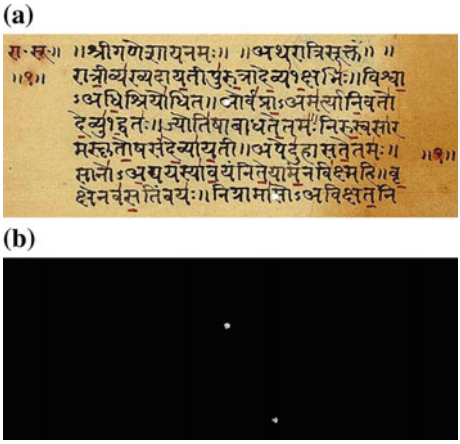


Fig. 2 **a** Deteriorated image of Devanagari manuscript. **b** Showing the positions of deterioration in above manuscript image

portion is visible. The deterioration holes in an image will be visible as white pixel and the rest of the image will be visible as black pixel image, by keeping some threshold value over an image in such a way, that it gets converted into a binary image. Figure 2 [9] shows the deterioration position as a white dot in an image and giving a pixel position of this position in (x, y) coordinates.

Finding the incomplete character

The next step after identifying the deterioration places in an image is to identify the incomplete character in an image. For identifying an incomplete character, we have to calculate the Euclidian distance for each character of Devanagari manuscript from data set. After identifying the Euclidian distance, we will have some threshold value for each image. Now, further we will apply a template matching, in which we will pass our deteriorated manuscript image and as a template, we will pass every character. Now, the matching in an image with a template image of a character will be performed using a previously identified threshold value. The above process will be performed at first place of deterioration, whose pixel position we have already identified.

Replacing the character

As we discussed previously in our second step, how can we get the information about the character which is incomplete and in first step we discussed about the location of deterioration in a manuscript image, with the help of these two steps it is quite easy for us to replace the incomplete character with the complete character that is taken from a data set. The data set is made up of these manuscript images. Now, we will pass that character from data set that will replace the incomplete character in a manuscript image.

3.3.2 Background Restoration Using Otsu Binarization Thresholding

The Otsu's method basically converts the grey-level image into binary image by performing clustering-based image thresholding that is automatically done. The Otsu binarization algorithm assumes that the image that is to be threshold should be a bimodal image means it must contain two classes of pixels, and then we can calculate the optimum threshold value through which we can separate these two classes in order to minimize the combined spread (intra-class variance). In Otsu's method, in order to minimize the intra-class variance, we need to find out a threshold value exhaustively. The intra-class variance is the variance within classes, which is defined as a weighted sum of variance of the two classes

$$\sigma_{\mathrm{w}}^2(\mathrm{t}) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t) \tag{1}$$

The probability of the two classes weighted as w_i, which is separated by a threshold t and the variances of these classes is σ_i^2.

Otsu shows that maximizing inter-class variance is same as minimizing the intra-class variance,

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = w_1(t)w_2(t)[\mu_1(t) - \mu_2(t)]^2 \tag{2}$$

This is expressed in terms of w_i class probabilities and class means μ_i.

The class probability $w_1(t)$ is computed from the histogram as t:

$$w_1(t) = \sum_0^t p(i) \tag{3}$$

While the class mean $\mu_1(\mathrm{t})$ is:

$$\mu_1(t) = \left[\sum_0^t p(i)x(i)\right]/w_1 \tag{4}$$

where $x(i)$ is the value at the centre of the ith histogram bin. Similarly, you can compute $w_2(t)$ and μ_i on the right-hand side of the histogram for bins greater than t.

Iteratively, we can compute the class probabilities and class means. This idea yields an effective algorithm.

Algorithm [2]

- Computation of histogram and probabilities on each intensity level.
- Initial values of $w_i(0)$ and $\mu_i(0)$ were set up.
- All possible thresholds values as $\mathrm{t} = 1 \ldots.$ maximum intensity were set up,
- Next is to update the values of w_i and μ_i,
- Now Computation of $\sigma_b^2(t)$ is done,
- Desired threshold corresponds to the maximum of $\sigma_b^2(t)$.

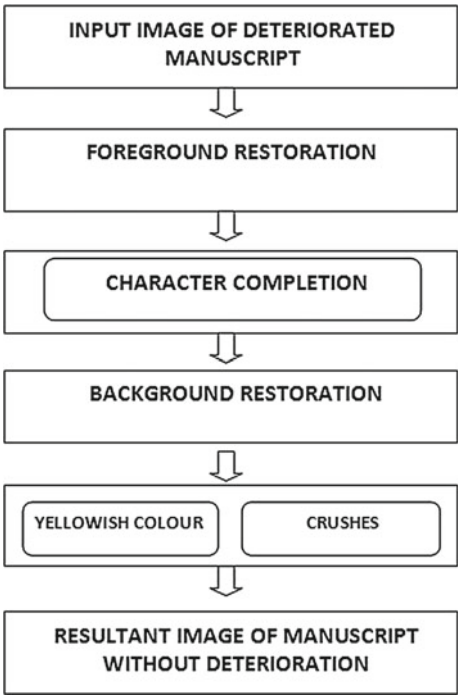


Fig. 3 Proposed model of framework

- You can compute two maxima (and two corresponding thresholds). $\sigma_{b1}^2(t)$ is the greater max and $\sigma_{b2}^2(t)$ is the greater or equal maximum.
- Desired threshold $= \frac{\text{Threshold}_1 + \text{Threshold}_2}{2}$

A trial and error method is used in order to know a value selected is good or not, by arbitrary selecting a value as a threshold value in a global thresholding technique. The Otsu binarization thresholding technique calculates automatically a threshold value from a histogram of an image. In simple words, it automatically calculates a threshold value from image histogram for a bimodal image. (For images which are not bimodal, binarization won't be accurate). We applied global thresholding for a value of 127.

We have already discussed the steps and technique in order to restore the old Devanagari manuscript, which have been deteriorated. The whole method for restoring the foreground and background deterioration in an image is shown in a proposed model. In the proposed model, we have taken deteriorated Devanagari manuscript image as an input. Firstly, the foreground restoration on an RGB manuscript image will be performed on one side and as an output we will get an RGB image free from foreground deterioration, which is having all complete characters. Now, we will focus on background restoration on an RGB input image. In the background restoration, by applying the above-discussed techniques, we will get an output image, which is free from yellowish background and crushes in an RGB image (Fig. 3).

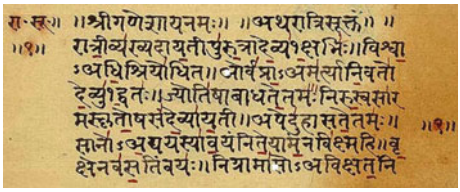


Fig. 4 Deteriorated image of Devanagari manuscript

4 Result

The results are evaluated on a data set that is taken from VRI [9]. The data set contains all characters that are made from manuscript images. There are some of the constraints while making a data set because the manuscripts are handwritten Devanagari manuscripts so every manuscript image is having a different data set, containing all characters of Devanagari. The result evaluated on the images by using the above-proposed approach, and it is working well on these images of Devanagari manuscripts for restoring the foreground as well as background of an image. The results are 80% accurate on foreground restoration and successfully restored background restoration also. There are some 4–5 characters, which are having biased nature, so the results are not very satisfactory for them. The results are shown below on a manuscript image (Fig. 4).

Foreground Restoration

See Fig. 5.

Background Restoration

See Fig. 6.

5 Conclusion

There are many existing techniques for text separation from foreground and background in order to restore both foreground/background of manuscripts digitally, but we aim to complete the incomplete characters in foreground of RGB image for restoring a Devanagari manuscript. The aim of proposed methodology is to restore both foreground and background of manuscript image. Our work mainly focuses on restoring the foreground of RGB image of manuscript.

The proposed methodology improves the quality of deteriorated manuscript by removing the yellowish background, crushes and completing each incomplete character in manuscript.

Fig. 5 Restoring the foreground of a manuscript image by completing the character in an image. **a** Completing the incomplete character at first position; **b** completing the incomplete character at second position

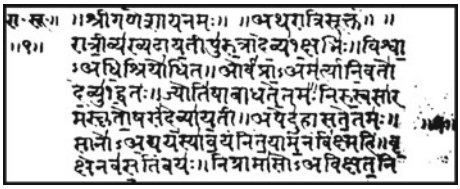


Fig. 6 Restoring the background of an image

6 Limitations

Few constraints of our work are:

- Only those characters can be corrected and completed where the deterioration rate is 50% or less than 50%.
- We can complete a single character at one position at a time.

The drawbacks of the proposed work are that you cannot restore complete manuscript at a time that is having multiple distortions; it can be only done one by one at each portion of it.

The research can be further extended to remove the above-mentioned constraints.

Acknowledgements I would like to give hearted thanks to my supervisor, Dr. Charul Bhatnagar, who has given me a lot of support, guidance and advice. I would like to express my special gratitude and thanks to the Head of Department, Computer Engineering and Applications, Prof. A. S. Jalal, for giving me attention and time whenever it was needed. I would like to express my gratitude to Vrindavan Research Institutes for providing the Devanagari manuscript images.

References

1. www.google.co.in
2. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. syst. Man Cybern. **9**(1), 62–66 (1979)
3. Kapur, J.N., Sahoo, P.K., Wong, A.K.C.: A new method for gray-level picture thresholding using the entropy of the histogram. Comput. Vis. Gr. Image Process. **29**, 73–285 (1985)
4. Kittler, J., Illingworth, J., Foglein, J., Paler, k.: An automatic thresholding algorithm and its performance. In: Proceedings of the 7th International Conference on Pattern Recognition, Montreal, pp. 287–289 (1984)
5. Chung, Y., Wang, J., Bailey, R., Chen, S., Chang, S.: A nonparametric blur measure based on edge analysis for image processing applications. In: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, pp. 80–89 (2004)
6. Boussellaa, W., Zahour, A., Alimi, A.: A methodology for the separation of foreground/background in arabic historical manuscripts using hybrid methods. In: Proceedings of the 2007 ACM symposium on applied computing, pp. 605–609 (2007)
7. Dimitrios, V., Ntogas, N., Ventza, M.: Digital restoration by denoising and binarization of historical manuscripts images (2012)
8. Ingole, Ketki R., Shandilya, V.K.: Image restoration of historical manuscripts. Int. J. Comput. Sci. Eng. Technol. **2**(4), 2229–3345 (2006)

9. Brown, B., Michael, S., Seales, W.B.: Image restoration of arbitrarily warped documents. In: Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1295–1306 (2004)
10. Boyatzis, S.C., Velivasaki, G., Malea, E.: A study of the deterioration of aged parchment marked with laboratory iron gall inks using FTIR-ATR spectroscopy and micro hot table. Herit. Sci. **4**(1), 13 (2016)
11. Sahoo, J., Mohanty, B.: Digitization of Indian manuscripts heritage: role of the national mission for manuscripts. IFLA J. **41**(3) pp. 237–250 (2015)
12. Seifi, L.: Digitization and Digital Preservation of Manuscripts and Access in Organization of Libraries, Museums and Documents Centre of Astan Quds Razavi Library in IRAN: A Case Study, pp. 281–284 (2011)
13. Manuscript Images: Vrindavan Research Institute, Mathura
14. Leydier, Y., Bourgeois, F.L., Emptoz, H.: (2004) Serialized K-mean for adaptive color image segmentation-application to document images and others. In: Proceedings of the 6th International Workshop on Document Analysis systems (DAS), Italy, LNCS, vol. 3163, pp. 252–263 (2004)
15. Garain, U., Paquet, T., Heutte, L.: On foreground-background separation in low quality color document images, In: Proceedings of the Eighth International Conference on Document Analysis and Recognition (2005)

Emotion Classification from Facial Expressions Using Cascaded Regression Trees and SVM

Manjot Singh Bilkhu, Samarth Gupta and Vinay K. Srivastava

Abstract Developing a robust automated system that can perform facial expression recognition is rather difficult. A system that performs these operations accurately and in real time would be a major step forward in achieving reliable human–computer interaction. In this paper, a new robust real-time algorithm is applied to face analysis to accurately predict emotions using facial expressions. A new facial pose estimation-based facial expression recognition system is introduced in this paper. A cascade of linear regressors is used to detect and track facial landmarks which will be used as features to train a classification algorithm. The landmark detector has been made to be robust to changes in the alignment of the faces. Support vector machine is used to classify emotions into one of the six emotions.

Keywords Support vector machines · Linear regressors · Facial landmarks Human–computer interaction

1 Introduction

Human–computer interaction [1] is the field of computer science that essentially attempts to bridge the gap between humans and machines. HCI aims to make machines perceptive enough to understand humans and what they are trying to communicate and then work accordingly. It is based on interdisciplinary researches that aim to implement knowledge from behavioral and social sciences into machines. In this context, the advantage of making machines to read human facial expressions is tremendous. A system involving such an analysis assumes that the face can be

M. S. Bilkhu (✉) · S. Gupta · V. K. Srivastava
Motilal Nehru National Institute of Technology Allahabad, Teliarganj, Allahabad, India
e-mail: manjotms10@gmail.com

S. Gupta
e-mail: samarthgupta95@gmail.com

V. K. Srivastava
e-mail: vinay@mnnit.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_44

accurately detected and tracked, the facial features can be precisely identified, and that the facial expressions, if any, can be precisely classified and interpreted.

The task of emotion recognition is particularly difficult for two reasons: the first being that there does not exist a large database of training images, and secondly, classifying emotions can be difficult in real time depending on whether the input image is static or a transition frame into a facial expression. In addition, accounting for variations in lighting and subject position in a non-laboratory environment is challenging. This paper discusses the development of a fully automated system to classify emotions based on facial expressions in real time. The system developed here will be independent of various external factors such as background scenery, lighting, pose, and orientation.

This paper discusses the system that we have developed to classify the six basic emotions on the basis of facial expressions with an accuracy as high as 90%. The system accomplishes this task in real time using a live feed through a webcam with very little delay. This paper discusses various algorithms used to design different modules of the system.

The main algorithm that has been discussed here is concerned with the detection of faces and important landmarks on them, i.e., pose estimation, using cascaded regression [2], and subsequently extracting the important features and descriptors from these landmarks. A novel approach for feature extraction using cascaded regression is also discussed. Cascaded regression methods have been compared with methods like active appearance models (AAMs) [3, 4] and appearance-based templates that emphasize on using Gabor wavelets [5, 6] or local binary patterns (LBPs) [7, 8] to encode the face and try to find the best match. All these methods are used for face alignment and pose estimation, and it has been shown that cascaded regression provides the best accuracy in detecting and tracking facial landmarks among all the algorithms that have been discussed in Sect. 3.

The rest of this paper is organized as follows. Section 2 provides information about the dataset that has been used to train the classifier. Section 3 talks about the related work that has already been done in this field. The algorithm used to detect facial landmarks and extract features out of those landmarks is detailed in Sect. 4. Section 5 discusses the system that has been developed to classify facial expressions and the results obtained with different classification algorithms—SVMs, neural networks, and logistic regression. Section 6 concludes this paper and discusses the future work that could be done in this domain.

2 Dataset

The data has been obtained with due permission from CK+database [9, 10]. The CK+expression database is widely used for research in facial image attribution and synthesis and for perception. Cohn-Kanade is available in two versions.

The older version [9] has about 486 sequences from 97 different subjects. Each sample gradually proceeds from a neutral expression to a fully encoded FACS label.

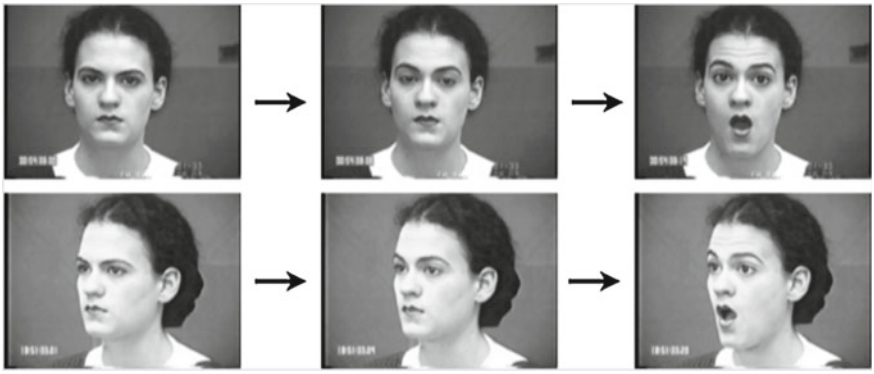


Fig. 1 Images from CK+ dataset

The latest version, referred to as CK+ [10], contains varied expressions and additional types of metadata, capturing spontaneous as well as intended expressions. The latest version sees an increase over the previous version both in terms of spontaneous as well as posed expressions. The expressions are fully FACS encoded like the previous release. In addition, validated emotion labels have been added to the metadata. For our training set, 46 images have been used for anger, 58 images for disgust, 69 images for happy, 309 images for neutral, and 83 images for surprise.

The data is preprocessed by normalizing the images to a size of 100×100 grayscale images. This reduces redundancy in the dataset as background information, when encoded, leads to huge losses during training and validation (Fig. 1).

3 Related Work

There has been a lot of work going on in facial expression analysis and related areas. In general, deep networks or specifically deep neural networks (DNNs) and convolutional neural networks (CNNs) have been flagged as breakthroughs. Jung and Lee [11] built both models and were successful in recognizing some emotions with good accuracies. However, the models failed to give good results in case of some emotions.

Lee and Jung [12] tried to improve the model by introducing a customized training step, which captured all image sequences of a subject. As expected, the system performed well after the introduction of this step, and there was a huge improvement as seen in the recognition rates of the emotions.

Recent progress has also highlighted the advantages of using deep learning specifically for feature extraction. The deep neural nets are known to perform extremely well when dealing with complex features. Each layer correspondingly picks up features in increasing order of complexity and tries to learn these features from the

training data. Chang and Hsu [13] implemented a CNN where they associated each layer with a specific feature extraction task, like attribute recognition, action unit detection, and arousal estimation, respectively.

4 Facial Landmark Detection and Feature Detection

Facial landmarks help us in coming up with a predicted shape of the face. A set of these keypoints also let us know about the alignment and orientation of the face. It is important to detect and track these keypoints in order to extract key information out of them, and ultimately, train our classifier.

We use the dataset mentioned in Sect. 2 and prepare our training set by extracting features from the dataset. Labels are assigned to each of the training sample processed according to the class it belongs to. This is essentially the basic concept involved in any supervised learning approach. A cascaded regression-based approach is not affected by the appearance of the face. Instead, it comes up with an estimate after several cascades where loss for each regressor is computed and minimized.

The following diagram explains how the model is trained to predict the shape. Feature vectors are extracted from the training phase and are used as training data where each feature vector has an associated true shape. The model is then tested against a subset of unseen training images (Fig. 2).

There are a lot of methods that detect and track facial keypoints based on appearance. Appearance-based template methods [14] compare the head of the detected face and try to find the closest match. They do so by matching the head with certain labeled examples. Some of these methods emphasize on using Gabor wavelets [5, 6] to encode the face and try to find the best match. While other methods use local binary

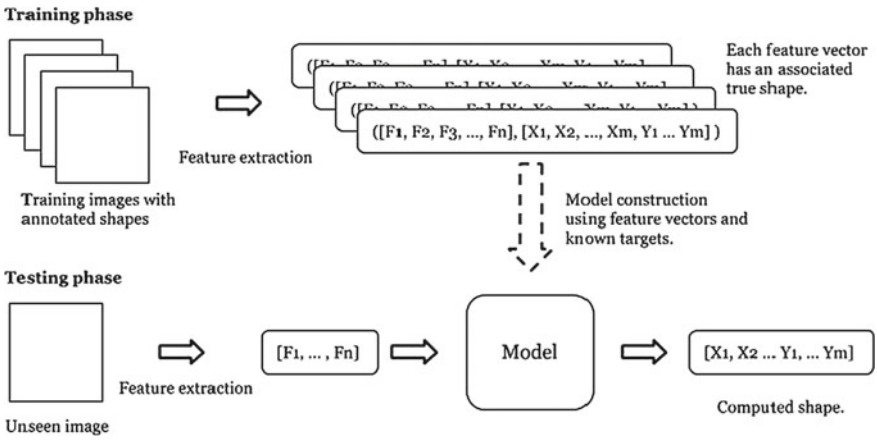


Fig. 2 Outline of face pose estimation

Fig. 3 Cascaded regression trees for facial landmark detection

patterns (LBPs) [7, 8] for facial representation for the purpose of facial expression recognition.

Generally, methods like active appearance models (AAMs) [3, 4] can be tuned to achieve robust performance. Conventionally, this was achieved by using some constrain solutions as training data. Also, AAMs use only a small number of parameters to describe an image.

The AAMs when used to detect and track landmarks in real time did not show very good results. This is because these models show huge variance as they are dependent upon the appearance of the face. These assume that the head has been detected and localized, but it is this very localization error that leads to a decline in the accuracy of face pose estimate. Also, adding more and more training examples to accommodate for the poor accuracy makes the system computationally expensive. Hence, regressor-based methods are widely preferred over AAM methods.

The most popular algorithms that use cascaded regression have an easily available open-source implementation. The state-of-the-art model that provides good results in real time is certainly the one proposed by Kazemi [2].

A cascaded regression [2, 15] architecture is constructed where each stage is an ensemble of regression trees (random forests). A decision tree is built to minimize the least squares error computed for alignment. S(t) describes the set of facial landmarks at stage t. Each stage takes the shape of the previous stage and adds a vector defined by the regression operation r_t. If I is the input Image, then this operation can be mathematically expressed as

$$S(t+1) = S(t) + r_t(I, S(t)) \tag{1}$$

To learn each regressor r_t, we use gradient tree boosting algorithm with sum of squares loss as defined in [2] to obtain the simulation results shown in Fig. 3.

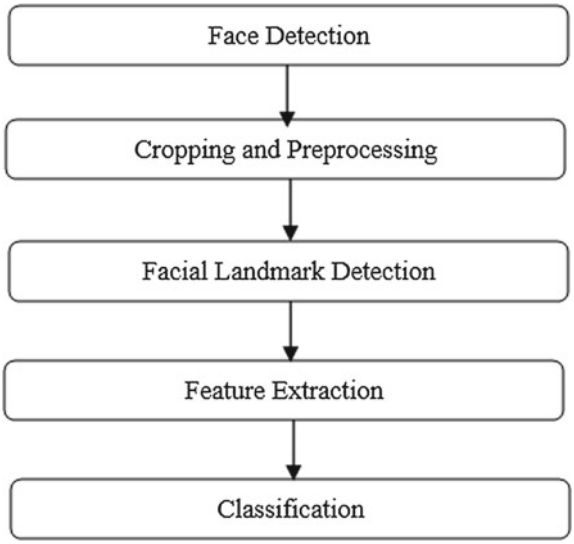


Fig. 4 Flowchart depicting the steps involved in the system

The mean of the detected landmarks is then computed. The topmost and bottommost landmarks of the nose are used to calculate the offset angle. Then, each landmark is subtracted from the mean. This incorporates geometric invariance into the system. The corresponding difference is computed to calculate the angle between each landmark and the mean. After subtracting the offset angle from these angles, these values can be used as features to train our classifier (Fig. 4).

5 Experiment and Results

The experiment that we have conducted to test the efficacy of our system involves the real-time detection of the faces of users on a live feed through a webcam. The captured images are then preprocessed to eliminate redundant information. To achieve this task, Haar cascade detection in OpenCV [16] was used to detect faces in the captured image. Object detection using Haar feature-based cascade classification or object detection is an approach, based on machine learning, in which a lot of positive and negative images are used as training data to train the cascade function. It is then used to detect objects (faces in this case) in test images. OpenCV comes with a trainer as well as detector. After successfully detecting the face from each image, the image of the face was cropped and saved as a new image (100×100) for each detected face. The new image is then converted into a grayscale image.

This step was followed by landmark detection by cascaded regression which has been explained in Sect. 3 of this paper. For this purpose, shape predictor method of Dlib was used. This method takes in an image region containing some object as input

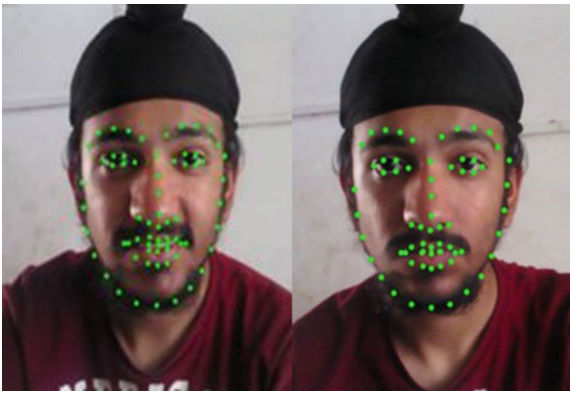


Fig. 5 Screenshot of real-time facial detection

and outputs a set of point coordinates that define the pose of the object. The output of the shape_predictor function used in this step is 68 (x, y) values containing the values of the various facial landmarks such as eyes, eyebrows, nose in the Cartesian plane (Fig. 5).

After calculating the 68 spatial coordinates corresponding to roughly the position of the nose tip in the face. After calculating the position of this central point, we calculate the distance and angle of each of the 68 points from this central point. There might be situations in which the orientation of the faces is tilted. This can be corrected if we work under the assumption that the bridge of the nose in most people is straight. This helps to accommodate rotational invariance in the proposed system and achieves good results in testing conditions where faces are tilted up to 30 degrees or less. To account for the tilted faces, the angles of the lines formed between the coordinates of the various landmarks and the coordinates of the mean point are offset by the angle of the bridge of the nose form. This rotates the entire vector array as if there was no rotation present at all. The process is illustrated in the image below. The image shows the line (in blue) formed between a coordinate and the mean coordinate which forms the angle α with the vertical. The other line (in red) shown follows the bridge of the nose. The angle β formed by the bridge of the nose with the vertical is the offset angle by which we will offset α. The features are selected as the angle between every detected point and the central point (mean point), after offsetting (Fig. 6).

The final data is obtained using the feature generation process described above and consists of M rows and 68 columns. M is the number of faces detected when using the system or the number of training examples used when training the classifier. There are 68 columns because each face corresponds to 68 features and each row corresponds to a single face. The training data is divided as 80% training data and 20% cross-validation data.

The system classifies facial expressions into six different basic emotions—happy, sad, neutral, disgust, fear, and surprise. The training data was divided into six different

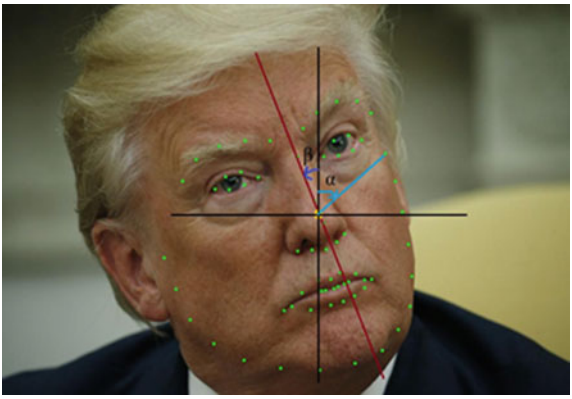


Fig. 6 Finding the angular offset β

Table 1 Different classifiers and their average accuracies

Classifier	Average accuracy (%)
Support vector machines	89
Logistic regression	80
Neural networks	77.06

folders, according to the emotion shown in the image, with each folder corresponding to one of the emotions listed above to train the classifier.

The system was tested with three different classifiers, namely logistic regression, neural networks, and support vector machines [17]. It was observed that support vector machines provided the best accuracy among the three classifiers followed by logistic regression and then by neural networks. In the code, a linear kernel was used to perform one versus all classification in SVMs. The neural network was built with one hidden layer and used with 10,000 iterations. Logistic regression was used with $\lambda = 0.5$. The accuracy of logistic regression increased with the number of iterations and reached a maximum at 5000 iterations. The accuracy of each of the three classifiers is shown in Table 1.

In this paper, a systematic approach to automated facial expression recognition has been presented. The paper discusses the method used to extract features from facial landmarks which have been obtained by using cascaded regression trees for face alignment and pose estimation. The paper also presents a comparison of various machine learning algorithms, such as logistic regression, neural networks, and support vector machines, that have been used for classification and discusses the accuracy obtained with each algorithm. Best results were obtained by using support vector machines resulting in accuracy as high as 89%.

6 Conclusion

The recognition system presented in this paper uses machine learning algorithms for classification and can be applied to the automated recognition of all images containing faces, provided a suitable training set is available. The expression recognition system developed here outputs results in real time, and the lag between the change in input and the change in output is negligible. The system is fully automated, and the results obtained are similar to or more accurate than those obtained with other facial expression recognition systems which did not employ cascaded regression trees for the purpose of feature generation.

In conclusion, we feel that while this system achieves a significant improvement over other models of real-time emotion recognition, it could be improved in the future by using 3D features instead of 2D features so as to enable it to classify a vast multitude of expressions instead of just classifying the six emotions that it currently does.

References

1. Bartlett, M.S., et al.: Real time face detection and facial expression recognition: development and applications to human computer interaction. In: Proceedings of the 2003 Conference on Computer Vision and Pattern Recognition Workshop, CVPRW'03 (2003)
2. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. Comput. Vis. Pattern Recogn. (CVPR) (2014)
3. Ishikawa, T., Baker, S., Matthews, I., Kanade, T.: Passive driver gaze tracking with active appearance models. In: Proceedings of the World Congress on Intelligent Transportation Systems (2004)
4. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Proceedings of the European Conference Computer Vision (1998)
5. Gu, W., Xiang, C., Venkatesh, Y., Huang, D., Lin, H.: Facial expression recognition using radial encoding of local gabor features and classifier synthesis. Pattern Recogn. **45**(1), 80–91 (2012)
6. Almaev, T.R., Valstar, M.F.: Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In: Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII), pp. 356–361 (2013)
7. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: a comprehensive study. Image Vis. Comput. **27**(6), 803–816 (2009)
8. Moore, S., Bowden, R.: Local binary patterns for multi-view facial expression recognition. Comput. Vis. Image Underst. **115**(4), 541–558 (2011)
9. Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, pp. 46–53 (2000)
10. Lucey, P., et al.: The extended Cohn-Kanade Dataset (CK+): a complete dataset for action unit and emotion-specified expression. In: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2010)
11. Jung, H., Lee, S., Park, S., Kim, B., Kim, J.: Development of deep learning-based facial expression recognition system. In: 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV) (2015)

12. Lee, I., Jung, H., Ahn, C.H., Seo, J., Kim, J., Kwon, O.: Real-time personalized facial expression recognition system based on deep learning. In: 2016 IEEE International Conference on Consumer Electronics (ICCE)
13. Chang, W.-Y., Hsu, S.-H., Chien, J.H.: FATAUVA-Net: An integrated deep learning framework for facial attribute recognition, action unit detection, and valence-arousal estimation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops
14. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models—their training and application. Comput. Vis. Image Underst. **61**, 38–59 (1995)
15. Doll_ar, P., Welinder P., Perona, P.: Cascaded pose regression. In: CVPR (2010)
16. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
17. Boser, B.E., Guyon, I.M., Vapnik, V.N.: Support vector machines are universally consistent. J. Complex. Jena, Germany 768–791 (2002)

Multi-layer Classification Approach for Online Handwritten Gujarati Character Recognition

Vishal A. Naik and Apurva A. Desai

Abstract In this paper, the authors present a multi-layer classification approach for online handwritten character recognition for the Gujarati characters. The Gujarati language consists of many confusing characters which lead to misclassification. Multi-layer classification technique is proposed to increase the accuracy of confusing characters. In the first layer of classification, SVM classifier with the polynomial kernel is used with all training data. If first layer classifier returns a character which can be confused with some characters than in the second layer, SVM with the linear kernel is used with confusing characters' training data. A hybrid feature set consisting zoning features and dominant point-based normalized chain code feature is used in both layers of classification. The system is trained using a data set of 2000 samples and tested by 200 different writers. The authors have achieved an average accuracy of 94.13% with an average processing time of 0.103 s per stroke.

Keywords Gujarati handwritten character recognition · Online handwritten character recognition (OHCR) · Handwritten character recognition (HCR)

1 Introduction

In the world of digitization, the demand for handheld devices is growing rapidly with time. There is a need of an easy and efficient tool to input data in the handheld devices. Mainly for Indian scripts, input using normal keyboard requires lots of efforts and time. Indian scripts have large character set with complexity of characters which makes an input to system difficult using a simple keyboard.

The handwritten character recognition system provides the best tool to input data in Indian scripts using natural handwriting. Handwritten character recognition is

V. A. Naik (✉) · A. A. Desai
Department of Computer Science, Veer Narmad South Gujarat University, Surat, Gujarat, India
e-mail: vishalkumar.naik@gmail.com

A. A. Desai
e-mail: aadesai@vnsgu.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_45

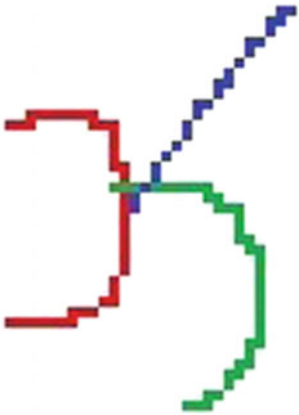


Fig. 1 Character ‘અ’

Fig. 2 Character ‘ઘ’ (left) and ‘ઘ’ (right)

getting more attention of researchers in the area of pattern matching because of availability and support of freehand natural handwriting tools. Small handheld devices have many advantages using online handwritten recognition over a simple keyboard.

This system captures the pen movement and converts it into pixel coordinate values. Meaning full features are extracted from raw pixel coordinates and passed to the classifier for recognition.

Gujarati is a native and official language of Indian state, Gujarat, which is an Indo-Aryan-type language. Gujarati has many similarities with Devanagari language. The Gujarati characters do not have Shirolekha over its characters. The Gujarati script character set consists of numerals, consonants, and vowels. Some characters require multiple strokes for a single character. For example, Fig. 1 illustrates character ‘અ’ which requires three different strokes. Some strokes are used in multiple characters. Some characters have high resemblance with other characters, like ‘ઘ’ and ‘ઘ’ (Fig. 2).

For Gujarati language, remarkable work is done in offline handwritten character recognition. Character segmentation method using radon transform is proposed in [1], and projection profile-based algorithm is proposed in [2]. Zone identification using Euclidean distance is proposed in [3].

Handwritten Gujarati numeral recognition is proposed in [4] using hybrid features, like a subdivision of skeletonized image and aspect ratio as a statistical approach. He has used k-NN as a classifier. He has achieved 82% accuracy. Similar work is proposed in [5] using four profile vector-based feature sets and used multi-layer feed forward neural network and achieved 82% accuracy.

Handwritten Gujarati character recognition is proposed in [6] using hybrid feature set that includes aspect ratio, extent, and 16 other features by dividing an image into 4×4 sub-images. He has used SVM with a polynomial kernel for classification and achieved 86.66% accuracy.

In [7], the authors have used low-level stroke features. They have used k-nearest neighbor classifier and achieved maximum 98.13% accuracy. In [8], the authors have used hidden Markov model (HMM) classifier. They have used zoning, open endpoints, and Zernike moments-based features for Gurmukhi characters. They have achieved an average of 93.50% accuracy. In [9], the authors have used artificial neural network using structural and wavelet transforms features for Kannada characters and achieved 91% accuracy. In [10], the authors have used weighted Euclidean distance technique using shadow-, centroid-, and distance-based features for Odia characters and achieved 87.6% accuracy.

Limited work is found for online handwritten character recognition for Gujarati characters. In [11], the authors have used k-nearest neighbors using low level and directional features and achieved 90% accuracy. In [12], the authors have used k-NN, SVM, and back-propagation NN using spatial and transform domain features and achieved 93.6% accuracy.

Notable work is done in online handwritten character recognition of other major Indian scripts like Assamese, Gurmukhi, Bangla, Marathi, and Hindi. In [13], the authors have combined HMM and SVM using resampled coordinate sequence, first- and second-order derivatives feature for Assamese characters, and achieved 96.17% accuracy. Similar work is presented in [14]; the authors have used two-stage classification approach using HMM and SVM using frequent count, used first- and second-order derivatives, slope, and baseline features, and achieved 95.1% accuracy. In [15], the authors have compared stroke-based and character-based method using HMM and found that character-based method gives a better result.

For Gurmukhi characters, in [16] the authors have presented work on a dominant point-based directional features using SVM and HMM classifiers and achieved 98.21 and 98.27% accuracies. Similar work is presented in [17]; the authors presented work on spatiotemporal and spectral features using k-NN, MLP, and SVM classifiers and achieved 89.35, 89.89, and 89.64%.

For Bangla characters, in [18] the authors have presented work on different geometric values as fuzzy features for recognition of Bangla character and achieved 77% accuracy. In [19], they have used structural and directional features per zone and slopes of dominant points feature per zone. They have used SVM classifier and achieved 87.48% and 85.10% using ZSD for Bangla and Devanagari and 92.48% and 90.63% using ZSDP, respectively.

For Marathi characters, in [20] the authors have presented work on rectangle histogram-oriented gradient features using SVM and FFANN classifiers and achieved 95.64 and 97.14%.

For Malayalam characters, in [21] the authors have presented work on accurate dominant points, start and end octants, and intersection features. They have compared k-NN, MLP, and SVM classifiers. They have achieved an accuracy of 90.39%, 93.17%, and 95.12%, respectively.

For Hindi characters, in [26] the authors have presented work on PCA and DTW using x–y coordinates as a feature. They have achieved an accuracy of 86% for PCA and 66% for DTW and average processing time of 0.8 s and 51 s, respectively.

For Chinese characters, in [22] the authors have presented work on a deep convolutional neural network (CNN) to identify Chinese characters and achieved 98.44% accuracy. For Arabic characters, in [23] the authors have presented work on different global and local features using SVM and achieved 92.43% accuracy. For Urdu characters, in [24] the authors have presented work on wavelet transform features using back-propagation neural network classier and achieved 91.3% accuracy.

The rest of the paper is divided into following sections. Section 2 describes preprocessing methods. Section 3 describes feature extraction methods. Section 4 describes classifiers. Section 5 describes results and discussion. Section 6 describes the conclusion of the paper.

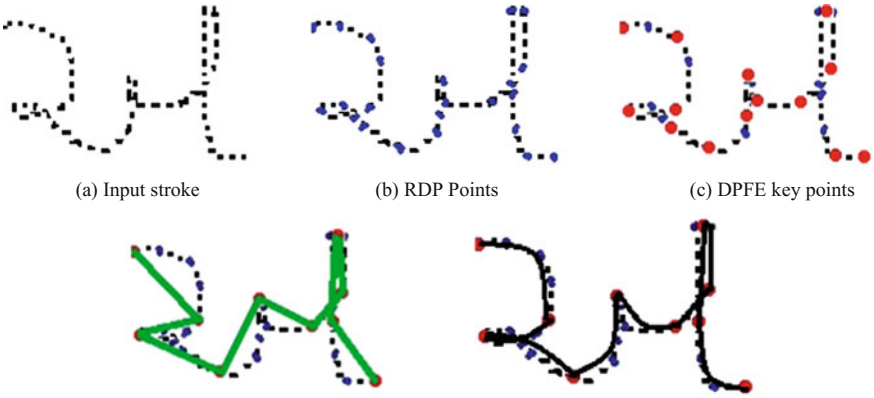
2 Preprocessing

The written stroke may vary from writer to writer in terms of size, style, etc. Size normalization and smoothing preprocessing methods are applied here. Variation in size of stroke leads to different feature values which decreases accuracy. Height and width of every stroke should be same to remove variation in size. We have used bilinear interpolation method to normalize a stroke. It will perform interpolation in both directions. Every stroke may contain some noise in it. To remove unwanted coordinates, we have used median filter smoothing technique. It is a nonlinear type of filtering technique.

3 Feature Extraction

Feature extraction is the most important step of the system because it extracts meaningful information from preprocessed raw coordinates. Raw coordinates contain plenty of data which should be filtered out to limited unique information which is known as a feature set. Different types of methods are used to extract a different type of features. Extracted set of features is given as an input to the classifier. We can divide features into global, local, structural, statistical, and hybrid features.

We have used different structural and statistical features as a hybrid feature set. The proposed feature set includes a percentage of active pixels in different zones, start and end zone, and dominant point-based normalized chain code features. Dominant point-based feature extraction (DPFE) algorithm [16] is used to find key points and direction between the key points of the stroke. The first set of dominant points are found using RDP algorithm [16]. Retrieved dominant points are further reduced to generate key points. Curve direction features can be extracted by measuring curve direction between key points. Curve directional information is represented as a chain



(a) Input stroke (b) RDP Points (c) DPFE key points

(d) Straight lines between key points (e) Curves between key points

Fig. 3 Dominant points-based features

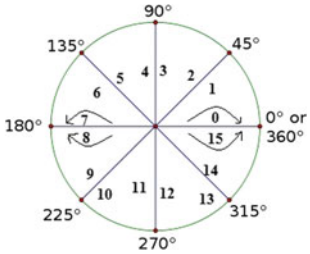


Fig. 4 Freeman chain code values for curve direction

code values. Chain code values are further normalized to make chain code starting point and rotation invariant. Figure 3 illustrates RDP points, DPFE points, straight lines between key points, and curves between key points for character 'અ'.

Input stroke is converted into RDP points which are further reduced to ten key points. Curve direction is measured for every pair of key points.

Figure 4 illustrates Freeman chain code values for different curve directions. These chain code values are used to represent particular curve direction numerically [25].

The Freeman chain code values for character 'અ' are 14, 10, 14, 1, 13, 1, 3, 11, 11, 13. To make it starting point invariant, the starting point should be redefined starting with minimum magnitude value in circular order.

The starting point normalized chain code values for character 'અ' are 1, 13, 1, 3, 11, 11, 13, 14, 10, 14. To make chain code rotation independent, we have to calculate the first difference of the chain code values. The first difference can be calculated by counting the number of directions that separate two adjacent elements of the code counter-clockwise. The rotation normalized chain code values for character 'અ' are 4, 12, 14, 8, 0, 13, 15, 4, 12.

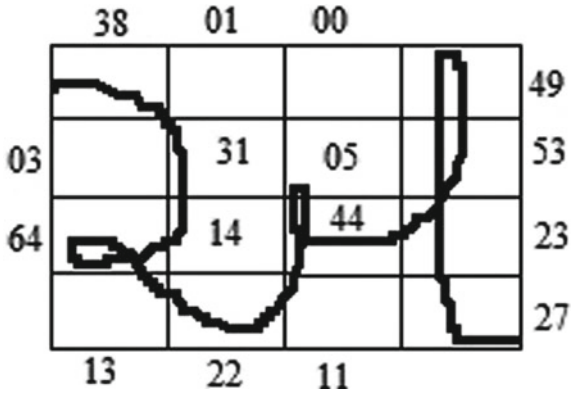


Fig. 5 Zoning of character "અ" into 16 equal zones

A local feature which describes different zones of a stroke, starting zone, and ending zone is used. In zoning, stroke is divided into 16 equal zones by considering. Percentage of a pixel in each 16 zone describes pixel distribution of a stroke in zones. Figure 5 illustrates zoning of character 'અ' with a percentage of active pixels in each zone.

4 Classification

The classification process is performed using different classifiers by inputting extracted feature set. We have used support vector machine (SVM) classifier. It is a supervised type of machine learning algorithm. Data is implicitly transformed into a higher dimension using different methods known as kernels. With the help of kernels, SVM can gain flexibility in selection of the threshold without occupying extra memory and time. SVM finds global optimum and unique solution due to quadratic programming. We have proposed multi-layer classification approach in which SVM with polynomial and linear kernel is used in a different layer.

A complete train set is used with first layer classifier which returns a class label. Based on the returned class label from first layer classifier, second layer classifier works. If the first layer classifier returns class label of any class which can be confused with other class, then based on that class label second layer classification can be performed with limited training data of confusing classes only. Figure 6 illustrates proposed block diagram.

Second layer classification is aimed to increase an accuracy of classifying confusing characters. An ambiguity of similar characters can be resolved using limited features and limited training data of confusing character classes.

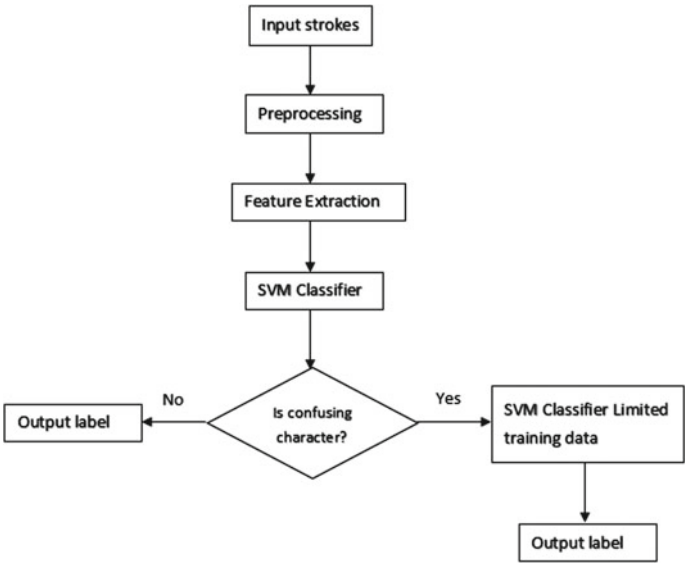


Fig. 6 Block diagram of the proposed system

5 Results and Discussion

Two types of training data set for a different layer of classification are used here. First layer classifier with complete training data set of around 2000 samples is collected from different writers. Second layer classifier uses training data set of confusing classes only. The proposed system is tested by 200 different writers with around 50 samples each. All the training data and testing data are captured using developed GUI system.

Table 1 shows a comparison between classifiers in terms of accuracy and an average processing time.

We have achieved highest accuracy of 94.13% using proposed multi-layer classification approach with an average processing time of 0.103 s per stroke. We have achieved an accuracy of 92.18% using a single SVM polynomial kernel with an aver-

Table 1 Comparison between classifiers

Classifier	No. of successful samples	Accuracy (%)	Avg. processing time (s)
SVM (polynomial)	9221	92.18	0.065
SVM (linear)	9064	90.63	0.046
Proposed multi-layer system	9429	94.13	0.103

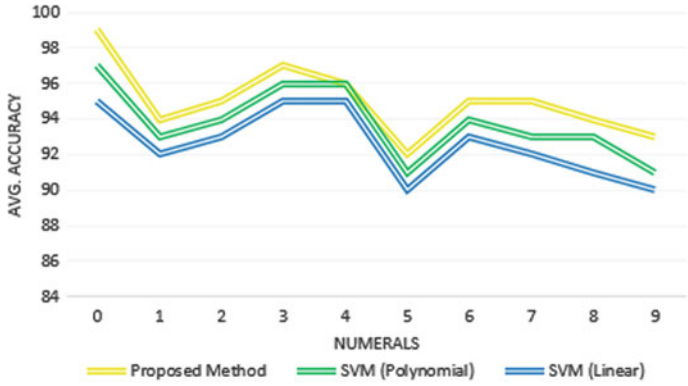


Fig. 7 Accuracy comparison of Gujarati numerals

age processing time of 0.065 seconds per stroke and 90.63% using a single SVM linear kernel with an average processing time of 0.046 seconds per stroke.

Figure 7 describes an accuracy comparison of Gujarati numerals in different classifiers. An average accuracy of Gujarati numerals for the proposed method is 95%, SVM with polynomial kernel is 93.8%, and SVM with a linear kernel is 92.6%. Figure 8 describes an accuracy comparison of a group of confusing character in different classifiers. An average accuracy of a group of confusing character for the proposed method is 92.14%, SVM with polynomial kernel is 88.42%, and SVM with a linear kernel is 87.57%. Figure 9 describes an accuracy comparison of a group of confusing character in different classifiers. An average accuracy of a group of confusing character for the proposed method is 92.2%, SVM with polynomial kernel is 89.8%, and SVM with a linear kernel is 89.8%. Figure 10 describes an accuracy comparison of a group of multi-stroke character in different classifiers. An average accuracy of a group of confusing character for the proposed method is 95%, SVM with polynomial kernel is 93.44%, and SVM with a linear kernel is 93%. Figure 11 describes an accuracy comparison of a group of multi-stroke character which requires more than two strokes in different classifiers. An average accuracy of a group of confusing character for the proposed method is 92.25%, SVM with polynomial kernel is 88.25%, and SVM with a linear kernel is 88.25%. Table 2 shows comparisons between major work done in different Indian languages. Comparison shows variety of classifiers and features for different Indian languages.

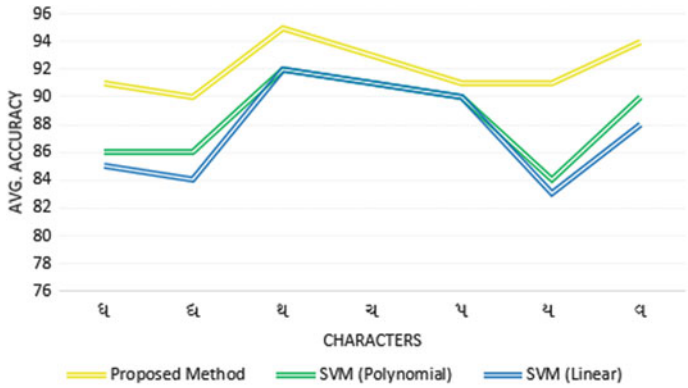


Fig. 8 Accuracy comparison of confusing characters Group 1

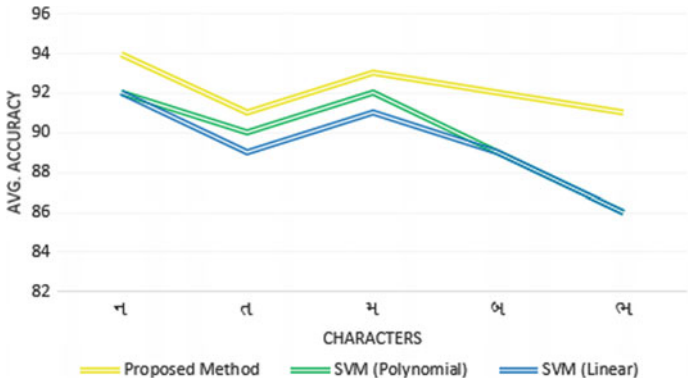


Fig. 9 Accuracy comparison of confusing characters Group 2

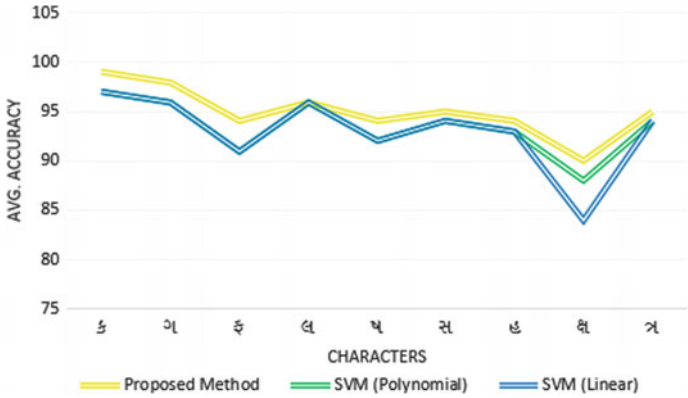


Fig. 10 Accuracy comparison of multi-stroke characters Group 1

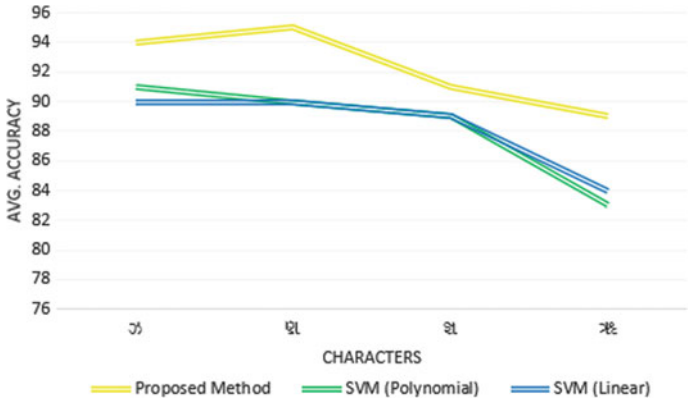


Fig. 11 Accuracy comparison of multi-stroke characters Group 2

Table 2 Comparison of different systems

Language	Classifier	Features	Accuracy (%)	Processing time (s)	Author(s)
Gujarati	Proposed multi-layer system	Zoning normalized chain code	94.13	0.103	The authors
Gujarati	k-NN	Low-level stroke directional	90	–	Gohel et al. [11]
Hindi	PCA	x–y coordinates	86	0.8	Prasad et al. [26]
Bangla	SVM	Zone-wise slopes of dominant points	92.48	–	Ghosh and Roy [19]
Assamese	Combined HMM and SVM	Resampled coordinates first- and second-order derivatives	96.17	–	Choudhury et al. [13]
Gurmukhi	HMM SVM	Dominant point-based directional features	98.27 98.21	–	Singh et al. [16]
Malayalam	k-NN MLP SVM	Accurate dominant points Aspect ratio Intersection	90.39 93.17 95.12	–	Bagu [21]

6 Conclusion

The authors proposed an algorithm for online handwritten Gujarati character recognition using hybrid features. The authors have compared proposed multi-layer classification approach with SVM (polynomial) and SVM (linear) classifiers. The authors have used a hybrid feature set with a training set of around 2000 samples. The proposed system is tested by 200 different users. The authors have achieved an accuracy of 94.13% and 0.103 s of average execution time per stroke. The limitation of the proposed system is that if first layer classifier fails then second layer classifier cannot give accurate results.

Acknowledgements The authors acknowledge the support of University Grants Commission (UGC), New Delhi, for this research work through project file no. F. 42-127/2013.

References

1. Desai, A.A.: Segmentation of characters from old typewritten documents using radon transform. Int. J. Comput. Appl. **37**(9), 10–15 (2012)
2. Patel, C., Desai, A.: Segmentation of text lines into words for Gujarati handwritten text. In: Proceedings of 2010 International Conference Signal Image Process, ICSIP, pp. 130–134 (2010)
3. Patel, C., Desai, A.: Zone identification for Gujarati handwritten word. In: Proceedings of 2nd International Conference Emerging Applications Information Technology, EAIT 2011, pp. 194–197 (2011)
4. Desai, A.A.: Handwritten Gujarati numeral optical character recognition using hybrid feature extraction technique. In: International Conference on Image Processing, Computer Vision, & Pattern Recognition, IPCV (2010)
5. Desai, A.A.: Gujarati handwritten numeral optical character reorganization through neural network. J. Pattern Recogn. **43**(7), 2582–2589 (2010)
6. Desai, A.A.: Support vector machine for identification of handwritten Gujarati alphabets using hybrid feature space. CSI Trans. ICT **2**(January), 235–241 (2015)
7. Goswami, M.M., Mitra, S.K.: Classification of printed Gujarati characters using low-level. ACM Trans. Asian Low-Resour. Lang. Inf. Process. **15**(4), 26 (2016), Artic. 25. https://doi.org/10.1145/2856105
8. Kumar, M., Jindal, M.K., Sharma, R.K.: A novel framework for grading of writers using offline Gurmukhi characters. In: Proceedings of National Academy Science India Section A—Physical Science, vol. 86, no. 3, pp. 405–415 (2016)
9. Pasha, S., Padma, M.C.: Handwritten Kannada character recognition using wavelet transform and structural features. In: 2015 International Conference on Emerging Resolution Electronic Computer Science Technology, pp. 346–351 (2015)
10. Rushiraj, I., Kundu, S., Ray, B.: Handwritten character recognition of Odia script. In: International Conference on Signal Processing, Communication, Power and Embedded System, SCOPES 2016—Proceedings, pp. 764–767 (2017)
11. Gohell, C.C., Goswam, M.M., Prajapate, Y.K.: On-line handwritten Gujarati character recognition using low level stroke. In: Third International Conference on Image Information Processing On-line, pp. 130–134 (2015)
12. Vyas, A.N., Goswami, M.M.: Classification of handwritten Gujarati numerals. In: 2015 International Conference Advanced Computer Communications and Informatics, ICACCI 2015, pp. 1231–1237 (2015)

13. Choudhury, H., Mandal, S., Devnath, S., Prasanna, S.R.M., Sundaram, S.: Combining HMM and SVM based stroke classifiers for online Assamese handwritten character recognition. In: 12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015, pp. 1–6 (2016)
14. Mandal, S., Choudhury, H., Prasanna, S.R.M., Sundaram, S.: Frequency count based two stage classification for online handwritten character recognition. In: 2016 International Conference on Signal Processing Communication SPCOM 2016 (2016)
15. Choudhury, H., Mandal, S., Devnath, S., Prasanna, S.R.M., Sundaram, S.: Comparison of Assamese character recognizer using stroke level and character level engines. In: 2015 21st National Conference on Communication NCC 2015 (2015)
16. Singh, S., Sharma, A., Chhabra, I.: A dominant points-based feature extraction approach to recognize online handwritten strokes. Int. J. Doc. Anal. Recognition **20**(1), 37–58 (2017)
17. Kaur, R., Singh, M.: Stroke based online handwritten Gurmukhi character recognition. In: 2016 International Conference on Advanced Computer Communiction Informatics, ICACCI 2016, pp. 598–601 (2016)
18. Chowdhury, K., Alam, L., Sarmin, S., Arefin, S., Hoque, M.M.: A fuzzy features based online handwritten Bangla word recognition framework. In: 2015 18th International Conference on Computer Information Technology ICCIT 2015, pp. 484–489 (2016)
19. Ghosh, R., Roy, P.P.: Study of two zone-based features for online Bengali and Devanagari character recognition. In: Proceedings of International Conference Document Analysis Recognition, ICDAR, vol. 2015, pp. 401–405 (2015)
20. Kamble, P.M., Hegadi, R.S.: Handwritten Marathi character recognition using R-HOG feature. Proc. Comput. Sci. **45**(C): 266–274 (2015)
21. Bagu, B.S.K.: Online Recognition of Malayalam Handwritten Scripts—A comparison using KNN, MLP and SVM. In: Proceedings of International Conference Advance Computer Communication Informatics, no. Sept. 21–24, 2016, pp. 1–5 (2016)
22. Zhang, S., Jin, L., Lin, L.: Discovering similar Chinese characters in online handwriting with deep convolutional neural networks. Int. J. Doc. Anal. Recogn. **19**(3), 237–252 (2016)
23. Nakkach, H., Hichri, S., Haboubi, S., Amiri, H.: Hybrid approach to features extraction for online arabic character recognition. In: Proceedings of Computer Graphical Imaging Visual New Technology Trends, CGiV 2016, pp. 253–258 (2016)
24. Safdar, Q.T.A., Khan, K.U.: Online Urdu handwritten character recognition: initial half form single stroke characters. In: Proceedings of 12th International Conference Fronteries Information Technology FIT 2014, pp. 292–297 (2015)
25. Bo YU X.Q., Guo, L., Tianyun, Z.: A curve matching algorithm based on Freeman chain code. In: International conference on Intelligent Computing and Intelligent Systems (ICIS), pp. 669–672 (2010)
26. Prasad, G.K., Khan, I., Chanukotimath, N.: On-line Hindi handwritten character recognition for mobile devices. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics—ICACCI'12, p. 1074 (2012)

A Local Information-Based Fuzzy C-Means for Brain MRI Segmentation

Anu Mangla and Chandan Singh

Abstract Segmentation of brain tissues from magnetic resonance imaging (MRI) is crucial for quantitative analysis of brain images. The fuzzy c-means (FCM) algorithm has proven to be an efficient approach for brain MRI segmentation. However, accurate segmentation results are hard to find due to the presence of noise. In this paper, we apply most commonly used local denoising filters to preprocess the image for obtaining better segmentation results. Then, we quantitatively compared various FCM-based state-of-the-art segmentation approaches with the proposed methods using Jaccard similarity (*JS*) and similarity index (ρ) on the simulated and clinical MRI. It is observed from the comparison results that the proposed methods provide more accurate segmentation results than the existing methods.

Keywords Segmentation · Denoising · FCM · Bilateral filter · Anisotropic diffusion filter · Total variation filter

1 Introduction

Segmentation of brain MRI means to specify each pixel or voxel to a specific tissue class. Brain MRI has significance in biomedical image processing. Effective extraction of useful information from MRI is essential for the treatment of diseases [1]. Normal brain MRI consists of tissues like cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM). Several automated algorithms have been developed for the segmentation of brain MRI. However, brain MRI segmentation is still a very challenging task, due to complicated brain structure, the presence of noise, and intensity inhomogeneity. FCM [2] is most widely used for brain MRI segmentation. FCM provides very high segmentation accuracy for uncorrupted MRI images. However, as medical images are often corrupted by Rician noise due to equipment and other environmental factors [3], therefore segmentation with FCM becomes problematic.

A. Mangla (✉) · C. Singh
Punjabi University, Patiala 147001, India
e-mail: anumangala90@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_46

Pham [4] include the spatial penalty on the membership function of original FCM to make the algorithm robust to noise. Ahmed et al. [5] proposed fuzzy c-means with spatial constraints (FCM_S) that performs the task of noisy MRI image segmentation and intensity inhomogeneity correction simultaneously. The spatial constraints in terms of neighborhood averaging term are added in the original FCM objective function. Chen and Zhang [6] proposed two variants of FCM_S, called FCM_S1 and FCM_S2. In FCM_S1 and FCM_S2, to reduce the effect of noise, the intensity value of each pixel in the image is first converted into filtered intensity value called mean and median filtered image, respectively. Clustering is then performed based on the filtered image. Thus, it reduces the computational cost and also enhances segmentation accuracy. Cai et al. [7] proposed the fast generalized fuzzy c-means algorithm (FGFCM), in which a new locality factor is defined for adding the spatial constraints. Krindis and Chatzis [8] proposed a fuzzy local information c-means (FLICM) that modifies the objective function of FCM by introducing local spatial and gray-level information to make the algorithm more robust to noise. The algorithm is free from any regularization parameter and achieves high segmentation accuracy. Adhikari et al. [9] introduced an efficient way to segment MRI brain images in the presence of noise. They proposed an algorithm that incorporates both the local and global membership values into the final membership values to provide noise insensitivity. Recently, Zhang et al. [10] proposed non-local fuzzy c-means (NLFCM) that extends FLICM algorithm by utilizing the non-local spatial information in place of local information used in FLICM to enhance the robustness to noise.

In this paper, we focus on most commonly used local denoising filters to preprocess the image for better segmentation accuracy. To accomplish this task, we use different denoising filters such as bilateral filter (BF), anisotropic diffusion (AD), and total variation (TV) filter to remove noise from the image and then used the filtered image in the clustering process. Bilateral filter has been very effective in the image restoration process to remove the image degradation due to noise and blur, while the total variation regularization and anisotropic filtering have been used as standard methods for image denoising because of their ability to keep prominent edges in the image. Keeping these aspects in view, we proposed bilateral-based fuzzy c-means (BFCM), anisotropic diffusion-based fuzzy c-means (ADFCM), and total variation-based fuzzy c-means (TVFCM). Detailed experiments are conducted on the simulated and real MRI for comparing the performance of the proposed methods.

The rest of the paper is organized as follows. Section 2 presents a brief overview of FCM-based segmentation approaches. Section 3 presents the proposed method. In Sect. 4, results obtained on simulated brain MRI, and real MRI, are presented, followed by conclusion in Sect. 5.

2 Backgrounds

2.1 FCM_S1 and FCM_S2

Chen and Zhang [6] incorporate the local spatial constraints into the standard FCM objective function by computing the pre-filtered image to make the algorithm robust to noise. The objective function of original FCM algorithm is modified as

$$J_s = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^m \left\| x_i - v_j \right\|^2 + \alpha \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^m \left\| \bar{x}_i - v_j \right\|^2 \tag{1}$$

where $\bar{x}$ represents the mean and median filtered image for FCM_S1 and FCM_S2, respectively. The parameter α controls the trade-off between the original image and the pre-filtered image. When the value of α is zero, the algorithm approaches to standard FCM. The membership matrix u_{ij} and cluster prototype v_j are updated as follows:

$$u_{ij} = \frac{\left(\left|\left|x_i - v_j\right|\right|^2 + \alpha \left\|\bar{x}_i - v_j\right\|^2\right)^{-\frac{1}{(m-1)}}}{\sum_{k=1}^{c}\left(\left|\left|x_i - v_k\right|\right|^2 + \alpha \left\|\bar{x}_i - v_k\right\|^2\right)^{-\frac{1}{(m-1)}}} \tag{2}$$

and

$$v_j = \frac{\sum_{i=1}^{N} u_{ij}^m (x_i + \alpha \bar{x}_i)}{(1+\alpha)\sum_{i=1}^{N} u_{ij}^m} \tag{3}$$

3 Proposed Method

The preprocessing of the image in FCM_S1 using mean filter removes the Gaussian noise, while in FCM_S2 the preprocessing using median filter removes the salt-and-pepper noise. Due to averaging of data in a given neighborhood, mean filters are well known for blurring the images. Although median filter does not provide excessive smoothing, it may provide distortion in the image by shifting the edges [11]. As MRI image contains fine structures, edges, and other image details in many parts of the image, the use of mean and median filter may remove the essential details from the image. For this purpose, we have used BF, AD, and TV filters that retain the fine structures in the image while removing the noise.

3.1 Bilateral Filter

Bilateral filter [12] is a nonlinear, edge-preserving, noise smoothing filter. The formulation is:

$$\bar{x}_i = \frac{\sum_{r \in N_i} w_r x_r}{\sum_{r \in N_i} w_r} \tag{4}$$

where

$$w_r = exp\left(-\frac{(i-r)^2}{2\sigma_D^2} - \frac{(x_i - x_r)^2}{2\sigma_R^2}\right) \tag{5}$$

where σ_D^2 and σ_R^2 are the smoothing parameters for domain and range filtering.

3.2 Anisotropic Filter

Perona and Malik [13] proposed the nonlinear anisotropic filter for image smoothing. It attempts to remove image noise without removing essential image parts. The formulation is [14]:

$$\bar{x}_i^{t+1} = \bar{x}_i^t + \frac{\lambda}{|N_i|} \sum_{r \varepsilon N_i} c\left(\left|\nabla x_{i,r}^t\right|\right) \nabla x_{i,r}^t \tag{6}$$

where

$$c(x) = \exp\left[-\left(\frac{x}{k}\right)^2\right] \tag{7}$$

where $c(x)$ is the diffusion coefficient, constant $\lambda \in (0, 1]$ determines the rate of diffusion, t denotes the iteration step, and k represents the gradient magnitude threshold between the image noise and edges.

3.3 Total Variation Filter

Rudin et al. [15] introduced total variation minimization algorithm. The main aim is to recover the original image $u(x)$ from the noisy image $v(x)$ as the solution of the minimization problem [16].

$$TVF_\lambda(v) = \arg\min TV(u) + \lambda \int |v(x) - u(x)|^2 dx \tag{8}$$

where $TV(u)$ denotes the total variation of u and λ is a given Lagrange multiplier and controls the degree of filtering of the obtained solution.

Thus, the algorithm for BFCM, ADFCM, and TVFCM is given as follows:

Step 1: Select the initial clusters C using $\left\{v_j^0 = (2j-1)/(2C)\right\}_{j=1}^{C}$, the size of the neighborhood window $n \times n$, the maximum number of iterations $maxiter$, fuzzification parameter m, and epsilon ε.

Step 2: Set the iteration counter $q = 0$.

Step 3: For BFCM, ADFCM, and TVFCM, compute the bilateral filtered image, anisotropic filtered image, and total variation filtered image using Eqs. (4), (6), and (8), respectively.

Step 4: Update u_{ij}^{q+1} using Eq. (2).

Step 5: Update v_j^{q+1} using Eq. (3).

Step 6: If $\left|v_j^{q+1} - v_j^q\right| < \varepsilon \; or \; q \geq maxiter$ then stop; otherwise, set $q = q + 1$ and go to step 4.

4 Experimental Results

In this section, we analyze the performance of various FCM-based clustering approaches. All the experiments are conducted on the available *BrainWeb* database [17] and clinical database available from *Internet Brain Segmentation Repository* (IBSR) [18]. Prior to segmentation, non-brain tissues are removed from the MRI. The ground truth images are available in both the databases to quantitatively evaluate the segmentation performance. The segmentation performance of different algorithms is evaluated using the Jaccard similarity (JS) [19] and similarity index ρ [9]. JS is the ratio between intersection and union of the segmented class S_j and ground truth class G_j. It is the indication of similarity between S_j and G_j and defined as:

$$JS(S_j, G_j) = \frac{|S_j \cap G_j|}{|S_j \cup G_j|} \tag{9}$$

Similarity index ρ is another important measure which not only considers similar pixels, but also the contribution of the dissimilar pixels toward its value. It provides the overall segmentation accuracy for all the classes. It is defined as:

$$\rho = \frac{1}{C}\sum_{j=1}^{C} \frac{2|S_j \cap G_j|}{|S_j| + |G_j|}. \tag{10}$$

The values of JS and ρ vary from 0 to 1, with a value near to one indicates more accurate segmentation. Each pixel is assigned to a class with maximum degree of membership to achieve the hard final segmentation. In our experiments, we have compared the performance of our proposed methods BFCM, ADFCM, and TVFCM with

the six existing FCM-based segmentation algorithms, including FCM [2], FCM_S [5], FCM_S1 [6], FCM_S2 [6], FGFCM [7], csFCM [9], and NLFCM [10]. The parameters for these algorithms are set as $n = 3$, $\epsilon = 0.0001$, $m = 2$, and $\alpha = 1.0$. The algorithms are implemented using Visual C++ 10.0 and MATLAB 7.10.0 (R2010a) under Microsoft Windows environment on a PC with 2.20 GHz CPU and 4 GB RAM.

4.1 Segmentation of Simulated MRI

In this section, we apply all the comparison algorithms to a T1-weighted 1 mm simulated brain MRI, having a size of $181 \times 217 \times 181$. The simulated brain slice, ground truth, and the segmentation results are shown in Fig. 1. Figure 1a, b shows the original image with 9% Rician noise and its corresponding ground truth image. The segmentation results of different algorithms are shown in Fig. 1c–l. It can be observed from Fig. 1k, l that both ADFCM and TVFCM are more robust to noise and provide relatively better segmentation performance. Moreover, the segmented images obtained from these two algorithms are comparable to the ground truth segmentation. To show the comparison of different algorithms quantitatively, JS values of CSF, GM, and WM are shown in Table 1. It is observed from Table 1 that the proposed ADFCM and TVFCM show higher segmentation accuracy than all the classical FCM-based segmentation algorithms. Among these two, TVFCM performs superior than ADFCM. To further analyze the performance of traditional and the proposed algorithms, we apply the segmentation algorithms on a brain slice with increasing Rician noise levels, such as 3%, 5%, 7%, and 9%. Figures 2, 3, and 4 show the segmentation results on different Rician noise levels for CSF, GM, and WM regions, respectively. We can analyze from Fig. 2 that in CSF region, at noise levels 3, 5, and 7%, csFCM has highest segmentation accuracy, while at 9% noise level, the proposed TVFCM has better segmentation performance. Figure 3 illustrates that in GM region, at noise levels 3 and 5%, FGFCM has highest segmentation performance, while at noise levels 7 and 9% the proposed BFCM, ADFCM, and TVFCM have shown superior segmentation results than all other algorithms. Figure 4 illustrates that in WM region, the proposed TVFCM shows the highest segmentation accuracy at each noise level.

To further assess the robustness of the proposed methods, we analyze the behavior of the proposed and the compared methods on Gaussian and salt-and-pepper noises on the simulated MRI images with different levels of noise. Table 2 demonstrates the performance of all the comparison and the proposed methods on various levels of Gaussian noise. It can be observed from Table 2 that the performance of ADFCM and TVFCM is better than all the comparison algorithms. Further, TVFCM provides better similarity index (ρ) values at each level of noise than ADFCM. Table 3 demonstrates the performance of all the comparison and the proposed methods on the salt-and-pepper noise. It can be observed from Table 3 that TVFCM provides highest

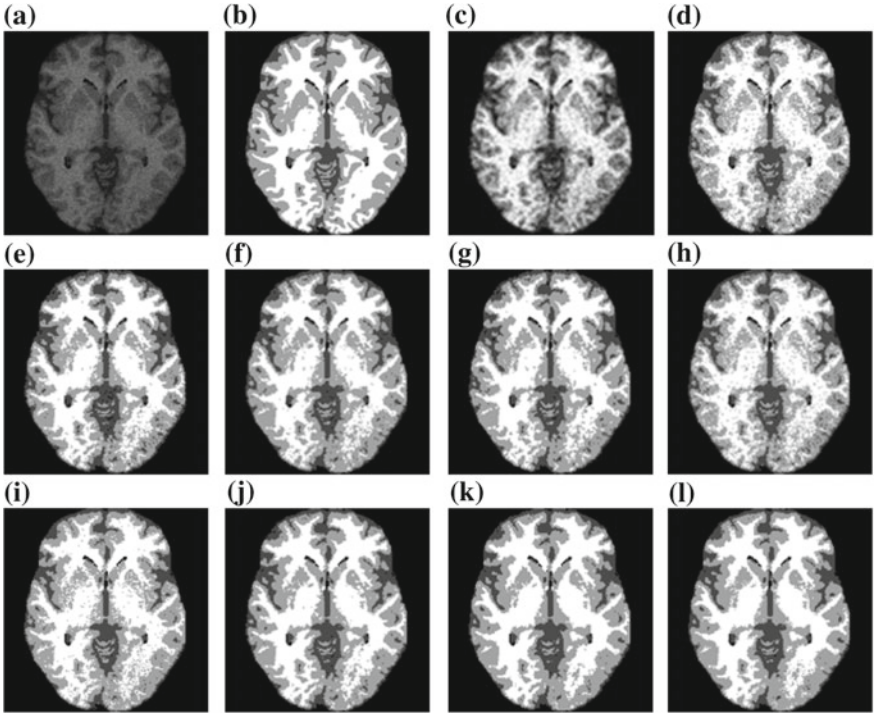


Fig. 1 Illustration of **a** original simulated T1-weighted brain MRI with 9% Rician noise, **b** show the segmentation ground truth, and the segmentation results obtained by applying **c** FCM, **d** FCM_S, **e** FCM_S1, **f** FCM_S2, **g** FGFCM, **h** csFCM, **i** NLFCM, **j** BFCM, **k** ADFCM, and **l** TVFCM

similarity index (ρ) values at each level of noise as compared to all the comparison methods.

4.2 Segmentation of Real MRI

In this section, we apply all the comparison algorithms to the T1-weighted real brain MRI available from Internet Brain Segmentation Repository (IBSR) [18], having a size of $256 \times 256 \times 128$. Like the segmentation accuracy reported by most of the authors, we do not present results for the CSF region because the ground truth images for real MRI include information only for the internal CSF, i.e., the ventricles, while the automated procedures also contain sulcal CSF. Thus, there is no proper base for the fair comparison of the CSF region. Both visual and quantitative results are analyzed to study the performance of both proposed and comparison methods. Figure 5a shows the T1-weighted original real MR image with 9% Rician noise, and Fig. 5b shows the corresponding ground truth segmentation available in the

Table 1 *JS* (%) values of CSF, GM, and WM regions of the ten algorithms applied on the simulated brain image (70th slice) with 9% Rician noise

Algorithm	CSF	GM	WM
FCM [2]	39.81	49.88	61.76
FCM_S [5]	61.87	68.19	76.54
FCM_S1 [6]	63.99	71.31	79.43
FCM_S2 [6]	71.43	74.88	80.08
FGFCM [7]	74.60	74.36	77.81
csFCM [9]	75.33	69.13	73.18
NLFCM [10]	74.91	74.90	79.53
BFCM (proposed)	71.69	74.69	79.87
ADFCM (proposed)	74.31	75.49	80.42
TVFCM (proposed)	76.02	76.34	80.89

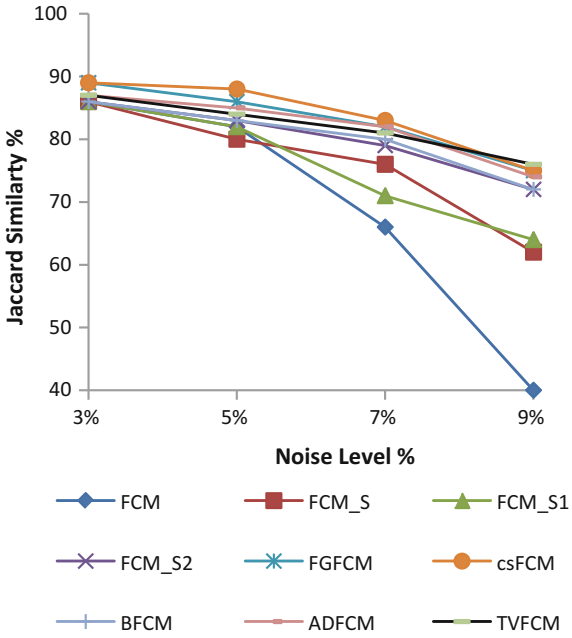


Fig. 2 Segmentation results for different noise levels on CSF

database. The segmentation results of different algorithms are shown in Fig. 5c–l. It is clearly observed from visual comparison that both proposed ADFCM and TVFCM approaches are much closer to the ground truth images than to those obtained by other methods.

We also investigate the performance of different segmentation algorithms quantitatively and measure the segmentation accuracy in terms of *JS* of GM and WM regions as shown in Table 4. It can be observed from Table 4 that the proposed ADFCM and

Table 2 ρ (%) values of segmentation results of the ten algorithms applied on simulated brain image (70th slice) corrupted with different levels of Gaussian noise

Algorithm (%)	FCM	FCM_S	FCM_S1	FCM_S2	FGFCM	csFCM	NLFCM	BFCM	ADFCM	TVFCM
Gaussian 3	79.37	83.45	88.83	89.31	89.56	87.59	88.83	88.91	89.62	90.19
Gaussian 5	62.35	79.50	86.94	87.07	87.39	81.83	86.38	87.03	87.76	87.85
Gaussian 7	56.63	76.20	85.22	84.57	85.91	76.92	84.24	85.32	85.48	86.43
Gaussian 9	53.01	72.30	83.49	82.26	84.16	71.67	77.11	83.63	84.67	85.52

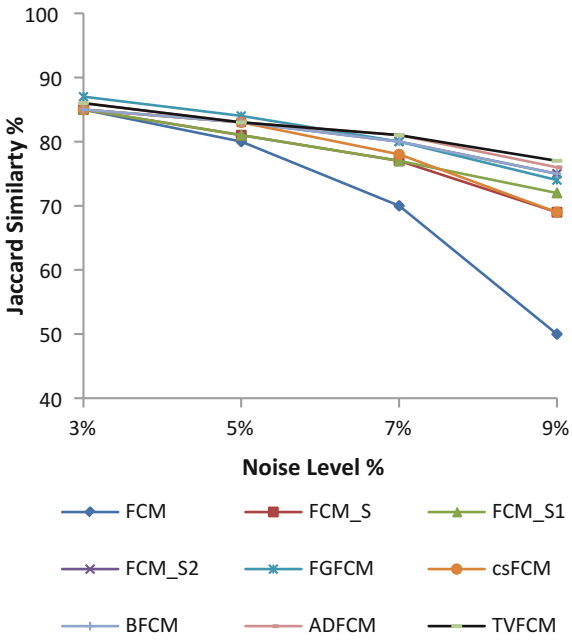


Fig. 3 Segmentation results for different noise levels on GM

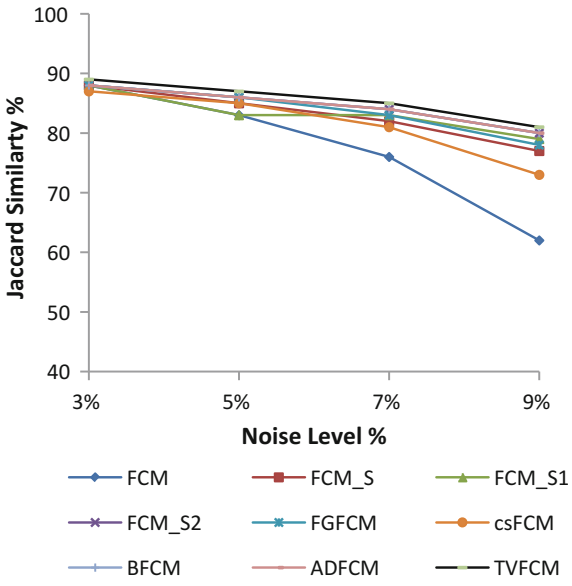


Fig. 4 Segmentation results for different noise levels on WM

TVFCM are more robust to noise and provide highest *JS* values than all the classical FCM-based methods. Further, TVFCM provides the highest segmentation accuracy as compared to the ADFCM.

Table 3 ρ (%) values of segmentation results of the ten algorithms applied on simulated brain image (70th slice) corrupted with different levels of salt-and-pepper noise

Algorithm (%)	FCM_S	FCM_S1	FCM_S2	FGFCM	NLFCM	BFCM	ADFCM	TVFCM
Salt and pepper 3	94.11	94.64	96.02	96.71	96.18	96.59	97.01	97.36
Salt and pepper 5	93.73	94.19	95.76	96.62	96.50	95.61	95.48	97.19
Salt and pepper 7	93.26	94.08	95.45	96.49	96.34	95.10	92.92	96.74
Salt and pepper 9	93.08	93.68	95.36	95.46	95.26	94.54	94.06	96.39

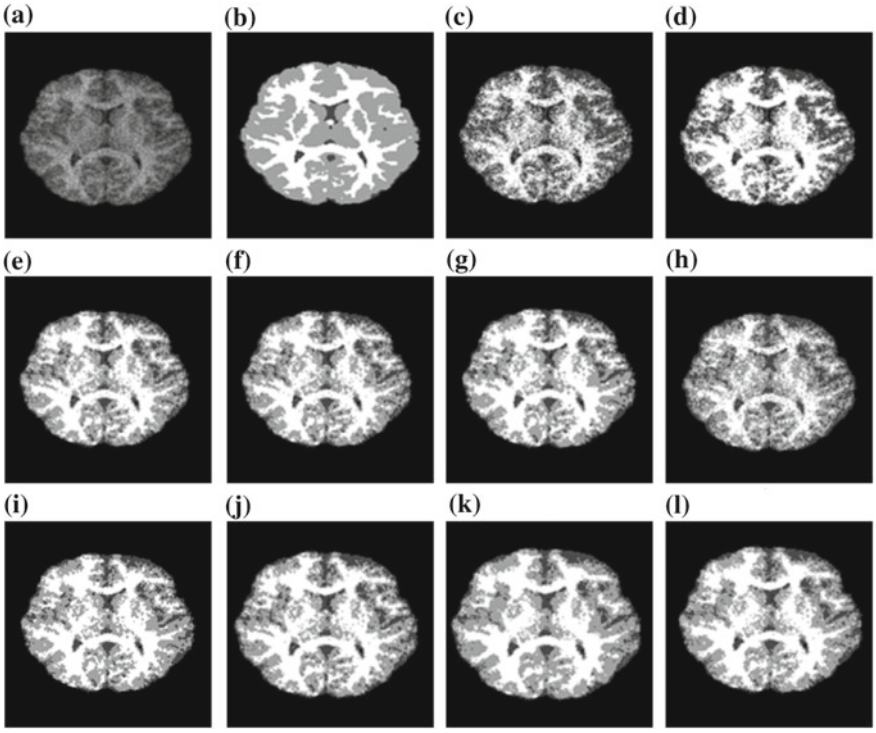


Fig. 5 Illustration of **a** original real T1-weighted brain MRI with 9% Rician noise, **b** show the segmentation ground truth, and the segmentation results obtained by applying **c** FCM, **d** FCM_S, **e** FCM_S1, **f** FCM_S2, **g** FGFCM, **h** csFCM, **i** NLFCM, **j** BFCM, **k** ADFCM, and **l** TVFCM

Table 4 JS (%) values of GM and WM regions of the ten algorithms applied on real T1-weighted brain image

Algorithm	GM	WM
FCM [2]	37.48	54.51
FCM_S [5]	40.82	66.17
FCM_S1 [6]	57.67	69.02
FCM_S2 [6]	60.61	68.83
FGFCM [7]	61.20	68.31
csFCM [9]	52.77	57.52
NLFCM [10]	58.36	68.15
BFCM (proposed)	59.80	69.80
ADFCM (proposed)	61.22	69.05
TVFCM (proposed)	61.91	69.84

5 Conclusion

In this paper, most commonly used local image denoising filters are used for computing the pre-filtered image to be used in the segmentation process. The segmentation performance of the proposed methods is compared with other FCM-based clustering approaches that are also based on local spatial constraints. The proposed methods are able to preserve fine structures of the image such as edges and corners and provide more robustness to noise than other existing FCM-based clustering approaches. Our comparison results on both simulated and real images show that the proposed BFCM, ADFCM, and TVFCM methods are capable of improving the segmentation accuracy as compared with other state-of-the-art methods.

Acknowledgements The authors would like to thank the University Grants Commission (UGC), New Delhi, India, for supporting the work by providing financial grants for the Major Research Project, entitled "Development of Efficient Techniques for Feature Extraction and Classification for Invariant Pattern Matching and Computer Vision Applications," vide its File No.: 43-275/2014(SR)

References

1. Guttmann, C.R., Jolesz, F.A., Kikinis, R., Killiany, R.J., Moss, M.B., Sandor, T., Albert, M.S.: White matter changes with normal aging. Neurology **50**(4), 972–978 (1998)
2. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the Fuzzy c-Means clustering algorithm. Comput. Geosci. **10**(2–3), 191–203 (1984)
3. Gravel, P., Beaudoin, G., Guise, J.A.D.: A method for modeling noise in medical images. IEEE Trans. Med. Imag. **23**(10), 1221–1232 (2004)
4. Pham, D.L.: Spatial models for fuzzy clustering. Comput. Vis. Imag. Understand. **84**(2), 285–297 (2001)

5. Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A., Moriarty, T.: A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. IEEE Trans Med. Imag. **21**(3), 193–199 (2002)
6. Chen, S., Zhang, D.: Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Trans. Syst. Man Cybern. **34**(4), 1907–1916 (2004)
7. Cai, W., Chen, S., Zhang, D.: Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. Pattern Recognit. **40**(3), 825–838 (2007)
8. Krinidis, S., Chatzis, V.: A robust fuzzy local information C-means clustering algorithm. IEEE Trans. Image Process. **19**(5), 1328–1337 (2010)
9. Adhikari, S.K., Sing, J.K., Basu, D.K., Nasipuri, M.: Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images. Appl. Soft Comput. **34**, 758–769 (2015)
10. Zhang, X., Sun, Y., Wang, G., et al.: Improved fuzzy clustering algorithm with non-local spatial information for image segmentation. Multimed. Tools Appl. **76**, 7869–7895 (2017)
11. Davies, E. R.: Image distortions produced by mean, median and mode filters, IEEE Proc. Vis. Image Signal Process **146**, 279–285 (1999)
12. Tomsi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of 6th International Conference on Computer Vision, pp. 839–846 (1998)
13. Perona, P., Malik, J.: Scale- space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intelligen. **12**(7), 629–639 (1990)
14. Yu, Y., Acton, S.T.: Speckle reducing anisotropic diffusion. IEEE Trans. Image Process. **11**(11), 1260–1270 (2002)
15. Rudin, L.I., Osher, S., Fatemi, E.: Non-Linear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)
16. Buades, A., Coll, B., Morel, J.: A nonlocal algorithm for image denoising. IEEE Comput. Soc. Conf. **2**, 60–65 (2005)
17. Brainweb Homepage. http://www.bic.mni.mcgill.ca/brainweb
18. IBSR Homepage. http://www.cma.mgh.harvard.edu/ibsr/
19. Ge, F., Wang, S., Liu, T.: Image-segmentation evaluation from the perspective of salient object extraction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (2006)

Analysis and Classification of Discriminative Region in Cognitive Functional MRI Data

J. Siva Ramakrishna and Hariharan Ramasangu

Abstract Feature engineering techniques such as feature selection and extraction dominate the process of cognitive state learning. The extraction of relevant features from high-dimensional multi-way functional MRI (fMRI) data is essential for the classification of a cognitive task. The dimensionality of fMRI influences the analysis of brain data. fMRI data is arranged as a number of voxels, region of interests (ROI) and snapshots. The extraction of a specific pattern of interest within the noisy components is a challenging task. In this paper, a tensor gradient-based feature extraction technique decomposes the multi-way fMRI data into a number of components. Voxel time series data from different ROIs has been used to find the region of discrimination. Clustering-based maximum margin feature selection method has been proposed to select the minimum number of voxels as attributes. The proposed techniques provide a better learning accuracy for the StarPlus fMRI data.

Keywords Signal processing · Pattern recognition · Functional MRI Classification · Clustering · Naive Bayes classifier · k-means clustering · Feature selection · Canonical poly-decomposition · Tensor factorization · Feature extraction · Discriminative region

J. Siva Ramakrishna is a research scholar at M. S. Ramaiah University of Applied Sciences, Bangalore, India.

J. S. Ramakrishna (✉)
Institute of Aeronautical Engineering, Hyderabad, India
e-mail: jsrkrishna3@gmail.com

H. Ramasangu (✉)
M. S. Ramaiah University of Applied Sciences, Bengaluru, India
e-mail: hariharan.ec.et@msruas.ac.in

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_47

1 Introduction

The advancement of noninvasive neuroimaging techniques improves the accuracy of detecting and decoding capacity of a human cognitive process. Functional brain imaging techniques provide real-time supervising of brain functioning. Over a decade, fMRI has become one of the mighty brain imaging tools for understanding and analysing the brain function. Functional MRI plays a significant part in human brain mapping to determine regions of the brain that are excited due to specific stimuli. Functional MRI offers a deep look into the human brain activation through a series of 3D brain images. The sequence of brain images is represented using blood-oxygen-level-dependent (BOLD) signal. These images leave deep insight into the underlying temporal and spatial changes of the human brain. Over the past few years, pattern recognition techniques such as classification and clustering have gained attraction in the fMRI data analysis. A large number of voxel information in fMRI data leads to over-fitting the classifier. The dimensionality reduction techniques provide appropriate attributes to the classifier, thereby improves the functioning of the classifier. In other words, different subjects in the data consist of different feature spaces.

In general, the neuroscientists mark the fMRI data into a different number of regions of interests (ROI). ROI analysis helps to explore each subject data. In the case of complicated designs like designs with multiple levels, it is difficult to extract a pattern of activity using whole-brain data. It is also useful to investigate the variation of signal levels across different variables at a specific region of interest. The other argument for the significance of ROI analysis is the type 1 error can be controlled by restricting the number of statistical tests using a fewer number of ROIs. fMRI records a combination of neural activation signals and several other noisy components. The extraction of a specific pattern of interest within the noisy components is a challenging task. Hence, techniques are needed to extract the actual underlying pattern of the source. Tensor decomposition techniques are used to extract the hidden patterns in the fMRI data [1]. Tensorization of fMRI data provides actual neural process within the multi-dimensional fMRI data. Tensor models help to find out the specific ROI that involved in the activation of a particular stimulus. The aim of the paper is to render discriminative ROI analysis for task-specific stimuli using tensor gradient technique. The rest of the paper is organized as follows: discussion about the related work is given in Sect. 2, the proposed tensor gradient-based discriminative ROI analysis is presented in Sect. 3, the proposed maximum margin feature selection technique is presented in Sect. 4, the benchmark fMRI data set used for the validation of the proposed method is explained in Sect. 5, the corresponding results for the proposed method is presented in Sect. 6, and conclusion is given in Sect. 7.

2 Related Work

ROI-based analysis of fMRI data helps to find the activation patterns of each ROI for a specific cognitive task. There are two ways to specify ROIs: one is structural, and the other is functional. Structural ROIs for each subject can be found on macro-anatomy. Functional ROIs are obtained by analysis of data from the same subject. The extraction of relevant features improves the classifier performance for cognitive task analysis. Neuroscientists mark the fMRI data into different into few numbers of ROIs. Each ROI comprises of task-specific information in terms of voxels. The classification of a cognitive state is performed using phase information extracted from Hilbert transform or Fourier transform. The phase information of the voxel time series is given as attributes to GNB classifier and attained learning accuracy of 93.7%. The learning accuracy of 99% is attained using SVM classifier [2]. Feature vectors are formed using the mean value of voxel time series. The property of BOLD signal such as delay in voxel time series is used to split the voxel time series into two halves. The classifier is built using mean values of first half of time series (m1) and the second half of time series (m2). Cognitive state classification is performed by combining both m1 and m2 as well as individual mean values of voxel time series. The results show that there is the significant improvement in classification accuracy for the feature vectors build through the second half of time series [3]. The classification of spatiotemporal StarPlus data has been performed using spiking neural network framework. Cognitive states are classified in four stage. In the first stage, data chunks are created from the fMRI data. In the second stages, NeuCube model is developed from the data chunks. NeuCube is basically a classifier for modelling spiking neural networks. The learning model consists of supervised and unsupervised stages for classification. NeuCube model adapts and addresses event representation technique to transform fMRI data into a series of spikes. The series of spikes are classified using spike pattern association and dynamic evoked spike neural network classifiers. The entire framework is optimized by modifying the parameter values in each iteration. The classification accuracy of 80–90% for the six subjects in StarPlus fMRI data is achieved using NeuCube framework [4].

The voxels selected from four significant ROIs are partitioned into ten number of clusters using hierarchical consensus clustering. Feature vectors are formed using randomly selected few number of voxels from each group. These feature vectors are used to build SVM classifier [5]. fMRI data can be defined as higher-order multi-dimensional tensor data, where a number of voxels, time instants, snapshots are considered as dimensions of the tensor. In the context of feature extraction, very few people have advocated factorization of fMRI data using tensor decomposition [6, 7]. Maximum margin feature selection framework and tensorization of fMRI data provide effective analysis for classification of a cognitive task. The main focus of the paper is to provide the brief analysis for classification of a cognitive task through feature selection and decomposition of fMRI data. The paper also discusses discriminative region of interests for cognitive state classification. We provide the analysis of a cognitive data with the help of a standard StarPlus fMRI data.

3 Proposed Tensor Decomposition-Based Discriminative Region of Interest Analysis for fMRI Data

Tensorization of high-dimensional data is an important step in functional MRI (fMRI) data analysis. Conventional methods to obtain lower-dimensional tensor data include matrix factorization such as applying tensor decomposition techniques like Tucker decomposition and canonical poly-decomposition (CP). CP decomposition is also popularly represented as CANDECOMP or parallel factorial decomposition (PARAFAC). The proposed discriminative region analysis for cognitive state classification is shown in Fig. 1. Functional MRI is a source for a large dimensional tensor data. Tensorization of fMRI data provides multiple factors for data analysis. The generated factors are used as features for learning the cognitive state. The proposed method is a six-step approach for cognitive state classification. The method aims to describe the significance of each ROI in terms of classification of a cognitive task.

Step 1: Selection of ROI is the first step in the proposed framework. In general, fMRI data is arranged as a fewer number of ROIs. Each ROI is defined in terms of a number of voxels and trials. As the proposed framework aims to determine discriminative regions for cognitive state classification, each ROI is selected independently and verified its significance for cognitive state classification.

Step 2: Function MRI data is an example for third-order array. The higher-order array consists of a number of voxels, time, trial as three components. In this step, the three-way data array has been converted a tensor of order 3.

Step 3: Fast alternative least square (ALS) is an improved version of ALS technique. Fast ALS technique has been employed on three-way cognitive tensor data to calculate CP gradients for the given tensor data. The three-way tensorized data into factors of R components. The fast ALS is a modified version of the CP ALS algorithm. CP fast ALS provides a second-order tensor with three factors.

$$X \in \Re^{J_1 \times J_2 \times J_3} \tag{1}$$

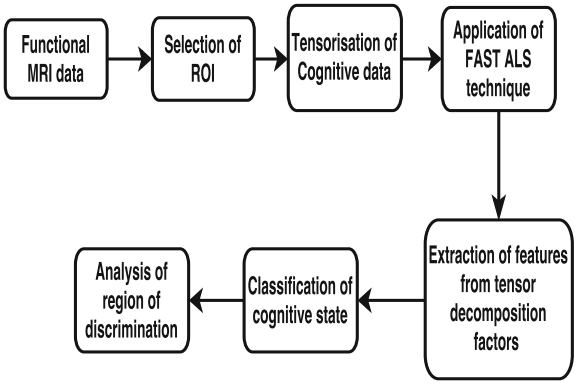


Fig. 1 Proposed tensor-based discriminative region analysis for fMRI data analysis

factorize X into 3 factors

$$A^{(3)} = [a_1^{(3)}, a_2^{(3)}, ...a_R^{(3)}] \tag{2}$$

$$A^{(3)} \in \Re^{J_n \times R}, [n = 1, 2, 3] \tag{3}$$

where R is a rank of a tensor

$$w^{(3)} \approx E_{(3)}(\bigodot_{k \neq 3} A^{(3)}) \tag{4}$$

$w^{(3)}$ is a CP gradient of tensor and $E_{(3)}$ is a $mod - 3$ unfolding error tensor.

Fast ALS technique performs tensor unfolding on the entries of the tensor matrices. In general, the entries of the tensor X are represented as a lengthy vector in the memory. In this point of view, tensor unfolding changes the length of the vectors as the changes in the entries increases. Tensor Box is used for tensorization of fMRI data.

Step 4: The CP gradients calculated in the previous step are extracted as features for cognitive state classification. In order to speed up the computation of canonical poly-decomposition gradients, a tensor is arranged in ascending order.

Step 5: The extracted features are given as input to the classifier. The classification of a cognitive task is performed in a leave-one-out fashion.

Step 6: In this step, we provide the ROI-based analysis for fMRI data through the results obtained in the previous step.

4 Proposed Maximum Margin Feature Selection Technique for fMRI Data Analysis

The proposed clustering-based maximum margin feature selection for cognitive state classification is shown in the Fig. 2. The framework is developed based on the selection of significant voxels as features for cognitive task classification. The significant and less number of voxels as attributes is given to the Naive Bayes classifier as input. The classification accuracy of six subjects in StarPlus data is analysed at different time instants. Clustering-based maximum margin feature selection algorithm is next described.

$$Y = [y_1, y_2, y_3, ..., y_n] \in \Re^{L \times n} \tag{5}$$

represents L number of features and n samples. Let f number of features be used to represent actual data. Where $f < L$,

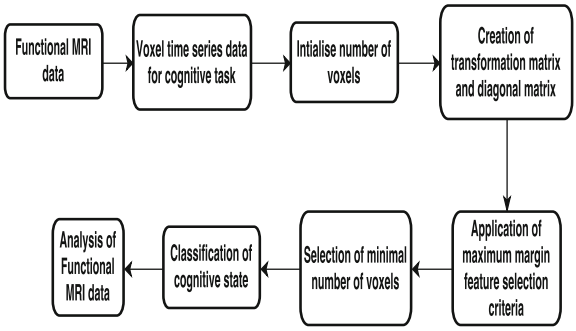


Fig. 2 Proposed maximum margin feature selection framework for fMRI data analysis

$W \in \Re^{L\times f}$ and $F \in \Re^{L\times L}$ represent the transformation matrix and diagonal matrix

$$f_{ii} = \frac{1}{(2\|w^i\|_2)}, i = 1, 2, 3, ...L \tag{6}$$

w^i is the ith row of W. $D \in \Re^{n\times k}$ is the indicator matrix, where k represents number of clusters. $D_{ij} = \frac{1}{\sqrt{l_j}}$, if y_i belongs to the ith cluster, else $D_{ij} = 0$; otherwise, l_j represents the amount of samples present in the jth cluster.

The proposed framework is discussed in the following seven steps:

Step 1: Voxel time series from a group of ROIs is chosen for the cognitive task classification of brain data.

Step 2: The required number of features from the original data set is initialized in the second step.

Step 3: The transformation matrix W and diagonal matrix F are created from the original data set.

Step 4: Maximum margin feature selection is applied on the matrices generated in the Step 3. The maximum margin technique for feature selection is explained as follows:

Maximum margin criteria (MMC):

MMC maximizes the mean margin among the classes in the protruded area [8]. Hence, feature selection criteria are defined as

$$C = \frac{1}{2}\sum_{a=1}^{k}\sum_{b=1}^{k} p_a p_b d(K_a, K_b) \tag{7}$$

K denotes the number of distinct classes, and p_a, p_b represents the prior probability of class a and class b. The interclass margin is given by

$$d(K_a, K_b) = d(m_a, m_b) - s(K_a) - s(K_b) \tag{8}$$

where $d(m_a, m_b)$ represents the interclass mean vector with m_a and m_b as the average vectors of the class K_a and K_b, respectively. $s(K_a)$ and $s(K_b)$ represent the covariance matrix of class K_a and K_b, respectively.

Step 5: Feature vectors are formed for the original data based on the indices of the features selected in Step 4.

Step 6: The feature vectors formed in Step 5 are applied as input to the classifier. Naive Bayes classifier used for cognitive task classification.

Step 7: The fMRI data is analysed based on learning accuracy obtained in the previous step.

5 Dataset

StarPlus data is a readily available fMRI data, for cognitive analysis of human brain. The data set is prepared by Carnegie Melon University [9]. The data set represents scanned fMRI data for human subjects. The data set depicts number observations and trials for six human subjects. During the scanned period, subjects are depicted by a picture ($\frac{*}{+}$) or ($\frac{+}{*}$) and then sentence: the $Star$ above the $Plus$ or the $Plus$ above the $Star$ during the first half of the experiment, and the order of the experiment is reversed for the other half of the trials. During the experiment, each stimulus is presented for 8 s, and 16 brain image sequences are captured within the 8 s time. The data set consists of a movie of activating levels within the brain, which are represented as voxels. Each 3-D fMRI image comprises of roughly 5000 number of voxels for each cognitive task.

The data set comprises of 54 trial information, which is experimented on human subjects. The subjects involve in cognitive task activity for 40 trials, and the subjects remain calm for the remaining trials. Hence, each cognitive task such as looking at a picture or saying a sentence has 40 trial data, for the analysis. In this framework, one trial data for picture task, sentence task is used for the cognitive state classification. The available StarPlus fMRI data set is already preprocessed to remove signal drifts and noise due to artifacts. Each subject in the original data set comprises of twenty-five ROIs. The voxels from seven significant ROIs—CALC, LT, LIPL, LTRIA, LIPS, LDLPFC and LOPER—are employed in the present work.

6 Discussion

Feature selection techniques play a significant role in the statistical analysis of data. It is a procedure of selecting most relevant attributes for effective model construction. Feature selection techniques are also recognized as attribute selection or variable selection methods. Feature selection methods are employed to simplify the training complexity of a classifier, reduce the dimensionality of the larger data set and reduce the over-fitting of a classifier.

6.1 Tensor Decomposition-Based Feature Extraction

Feature extraction techniques play a significant role in the statistical analysis of data. It is a procedure of extracting most relevant attributes for effective model construction. Feature extraction techniques are also recognized as feature engineering methods. Feature extraction methods are employed for training a classifier, for reducing the dimensionality of the larger data set and for reducing the over-fitting of a classifier. Tensorization of fMRI data decomposes the voxel time series data into multiple factors. Tensorization has performed over six subjects in StarPlus fMRI data. Each subject in StarPlus fMRI data comprises of approximately 5000 number of voxels and 25 number of ROIs. Cognitive state classification is performed using seven significant ROIs represented in the StarPlus data set. The ROIs used for the analysis include CALC, LT, LIPL, LTRIA, LIPS, LDLPFC and LOPER. Three-way tensor for subject 05710 is represented as $1883 \times 16 \times 40$, where 1883 denotes the number of voxels, 16 is the number of time points, and 40 is the number of trials. Similarly, for the subjects 04820, 04799, 04847, 05675, and 05680, the 3-way tensor is represented as $1888 \times 16 \times 40$, $1874 \times 16 \times 40$, $1715 \times 16 \times 40$, $2239 \times 16 \times 40$, $2230 \times 16 \times 40$ respectively. Tensor gradients are computed using a fast ALS algorithm. The features are extracted from the tensor gradients of voxel data. Tensor gradients, of different ROIs, are used to build Naive Bayes (NB) classifier. The aimed method is used to discriminate the ROIs in StarPlus fMRI data. The classification accuracy for the tensor gradient features of each ROI is compared with the Fisher discriminant ratio (FDR) [10] and statistics of split time series [3] methods. The classification accuracy comparison for the three methods is shown in the Tables 1, 2, 3 and 4.

Except for the ROI-CALC, the proposed tensor gradient-based feature extraction furnishes good learning accuracy for the remaining six noted ROIs. Among the seven significant ROIs, the features extracted from ROI-LT leaves a very good average classification accuracy of 99.37% for GNB classifier. Tables 5 and 6 present the classification accuracy for the features extracted from a combination of seven ROIs. The obtained results are compared with the other existing classification techniques such as linear functional approach [11], NeuCube [4], wrapper, embedded [12], Fisher discriminant ratio [10], clustering-classifier hybrid [5] approaches. The proposed tensor gradient-based feature extraction achieves an average classification accuracy of 96.4%, which is slightly better than the FDR method.

6.2 Maximum Margin Feature Selection

Unsupervised learning techniques employed on fMRI data partition the data into several groups. In this paper, k-means clustering with maximum margin via sparse constraints is used to fix the number of most relevant attributes for building an optimal model for cognitive state classification. The performance of the method has

Table 1 Classification accuracy for the features extracted from CALC

Subject	Method		
	Fisher discriminant	Split time series	Proposed method
04799	86	98.7	78.7
04820	71.2	88.7	90
04847	93	100	90
05675	86	97.5	93.7
05680	78	100	53.7
05710	88	98.7	47.5
Average	83.6	97.2	75.6

Table 2 Classification accuracy for the features extracted from LIPL

Subject	Method		
	Fisher discriminant	Split time series	Proposed method
04799	58	53.7	96.2
04820	65	50	92.5
04847	68	62.5	63
05675	56	53.7	100
05680	55	48.7	71.2
05710	61	52.5	70
Average	65.5	53.5	82

Table 3 Classification accuracy for the features extracted from LTRIA

Subject	Method		
	Fisher discriminant	Split time series	Proposed method
04799	67	68.7	78.7
04820	62.5	52.5	86.2
04847	58	36.2	65
05675	57	63.7	61.2
05680	61	60	65
05710	70	63.7	92.5
Average	62.5	53.5	74.7

been examined using StarPlus fMRI data for six subjects. Few numbers of attributes are selected for cognitive state classification using the advised method. Initially, the most important twenty feature indices are selected for one trial data, and the selected feature is extracted from each trial data to train the classifier.

Gaussian Naive Bayes (GNB) classifier is used for cognitive state classification. GNB belongs to the family of probabilistic classifiers. It operates on the principle of Bayes theorem with firm independency among features. Table 1 presents the average classification accuracy for six subjects in the StarPlus data set.

Table 4 Classification accuracy for the features extracted from LIPS

Subject	Method		
	Fisher discriminant	Split time series	Proposed method
04799	52	60	96.2
04820	60	73.7	82.5
04847	85	65	91.2
05675	63	83.7	100
05680	72	58.7	98.7
05710	58	65	83.7
Average	65	67.6	92

Table 5 Learning accuracy for the features extracted from seven ROIs

Subject	Method		
	Fisher discriminant	Split time series	Proposed method
04799	80	90	48
04820	95	90	86
04847	85	90	80
05675	95	80	90
05680	70	90	80
05710	90	90	91
Average	85.8	88.3	79.1

Table 6 Learning accuracy for the features extracted from seven ROIs

Subject	Method		
	Fisher discriminant	Split time series	Proposed method
04799	100	79.9	96.2
04820	96.2	74.8	83.7
04847	100	98.4	98.7
05675	93.7	92.8	100
05680	90	89	100
05710	95	94.2	100
Average	95.8	88	96.4

Since each subject has 40 samples for picture task, 40 samples for sentence task leave-one-out classification is performed over 80 samples of the data. The classifier is trained with 79 samples, and remaining one sample is used as test data for learning the class of test data.

The experimental results presented in Tables 7, 8, 9, 10, 11 and 12 show the comparison of classification accuracy for features selected using hierarchical consensus clustering (HCC) [5] and linear functional approach [11]. The results show phenomenal improvement in the classification accuracy for the proposed features selection

Table 7 Learning accuracy comparison for the subject 04799

Method	% Accuracy	# Voxels
Linear functional approach	80	846
Hierarchical consensus clustering	79.97	535
Proposed method	100	40

Table 8 Learning accuracy comparison for the subject 04820

Method	% Accuracy	# Voxels
Linear functional approach	95	728
Hierarchical consensus clustering	74.87	595
Proposed method	100	40

Table 9 Learning accuracy comparison for the subject 04847

Method	% Accuracy	# Voxels
Linear functional approach	85	885
Hierarchical consensus clustering	98.45	174
Proposed method	100	40

Table 10 Learning accuracy comparison for the subject 05675

Method	% Accuracy	# Voxels
Linear functional approach	95	1120
Hierarchical consensus clustering	92.87	580
Proposed method	100	40

Table 11 Learning accuracy comparison for the subject 05680

Method	% Accuracy	# Voxels
Linear functional approach	70	1051
Hierarchical consensus clustering	89	520
Proposed method	100	40

Table 12 Learning accuracy comparison for the subject 05710

Method	% Accuracy	# Voxels
Linear functional approach	90	810
Hierarchical consensus clustering	94.25	610
Proposed method	100	40

method compared to HCC and linear functional approach. The proposed method attains 100% accuracy for all the six subjects in StarPlus fMRI data set with 40 voxels.

Maximum margin feature selection has the advantage of selecting relevant features for classification and achieves 100% accuracy with minimum number of voxels. All the experimental simulations are performed on MATLAB 7.1 using Windows operating system with 10 GB RAM.

7 Conclusion

A novel feature engineering method for the classification of mind state has been proposed. The discriminative region in fMRI data is identified using a tensor gradient-based feature extraction technique. The selection of relevant voxels as attributes is accomplished through a new clustering-based maximum margin via sparse constraints feature selection method. The clustering technique considered is k-means, and L2,1-norm used as a minimization technique. The obtained results for various experiments are compared with other existing techniques. Overall, the two proposed methods achieve a good average accuracy for the six subjects in the StarPlus data set. The suggested methods are capable of producing significantly more accurate classification results compared to other methods.

References

1. Hunyadi, B., Dupont, P., Van Paesschen, W., Van Huffel, S.: Tensor decompositions and data fusion in epileptic electroencephalography and functional magnetic resonance imaging data. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 7, no. 1 (2017)
2. Ramasangu, H., Sinha, N.: Cognitive state classification using transformed fMRI data. In: 2014 International Conference on Signal Processing and Communications (SPCOM), p. 15. IEEE (2014)
3. Ramakrishna, J.S., Ramasangu, H.: Classification of cognitive state using statistics of split time series. In: IEEE Annual India Conference (INDICON), p. 15. IEEE (2016)
4. Murli, N., Kasabov, N., Handaga, B.: Classification of fMRI. Springer pp. 421–428 (2014)

5. Ramakrishna, J.S., Ramasangu, H.: Cognitive state classification using clustering-classifier hybrid method. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1880–1885. IEEE (2016)
6. Allen, G.I.: Regularized tensor factorizations and higher-order principal components analysis. arXiv:1202.2476 (2012)
7. Phan, A.-H.: P. Tichavsk'y, and A. Cichocki, Tensor deflation for candecomp/parafacpart i: alternating subspace update algorithm. IEEE Trans. Signal Process. **63**(22), 5924–5938 (2015)
8. Yang, S., Hou, C., Nie, F., Wu, Y.: Unsupervised maximum margin feature selection via l2, 1-norm minimization. Neur. Comput. Appl. **21**(7), 1791–1799 (2012)
9. Just, M., Mitchell, T.: Starplus FMRI data. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www (2001)
10. Do, L.N., Yang, H.J.: Classification of cognitive states from fmri data using fisher discriminant ratio and regions of interest. Int. J. Cont. **8**(4), 56–63 (2012)
11. Chen, X., Xu, M., Wu, W.B.: Regularized estimation of linear functionals for high-dimensional time series. arXiv:1506.03832 (2015)
12. Fan, M., Chou, C.-A.: Exploring stability-based voxel selection methods in mvpa using cognitive neuroimaging data: a comprehensive study. Brain Informat. **3**(3), 193–203 (2016)

Face Image Super-Resolution Using Differential Evolutionary Algorithm

Shyam Singh Rajput, K. V. Arya and Vijay Kumar Bohat

Abstract Image resolution enhancement plays the vital role in numerous real-world applications. In this work, a novel attempt toward the enhancement of the quality and resolution of the input low-resolution images using evolutionary algorithms has been made. The proposed image super-resolution (ISR) framework utilizes the differential evolutionary (DE) algorithm for minimizing least square estimation problem to obtaining the optimal reconstruction weights. To show the super-resolution capability of the proposed ISR framework, experimental tests are performed on widely used public FEI dataset. The simulation outcomes exhibit that the proposed framework outperformed the two very popular methods.

Keywords Differential evolutionary algorithm · Super-resolution · Image reconstruction

1 Introduction

As the technology growing, the demand for high-quality and high-resolution (HQHR) images and videos is also increasing. The mechanism used to enhance these factors in the low-quality and low-resolution (LQLR) image is called image super-resolution (ISR). The ISR mechanism is getting considerable attention in recent years because of its wide applicability in the field of image and vision computing, e.g., computer–human interaction, face recognition, video surveillance, high-resolution satellite imaging system, video conferencing, magnetic resonance imaging system.

S. S. Rajput (✉) · K. V. Arya · V. K. Bohat
Multimedia and Information Security Research Group, ABV—Indian Institute of Information Technology & Management, Gwalior 474015, India
e-mail: ershyamrajput@gmail.com

K. V. Arya
e-mail: kvarya@iiitm.ac.in

V. K. Bohat
e-mail: vijay.bohat@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_48

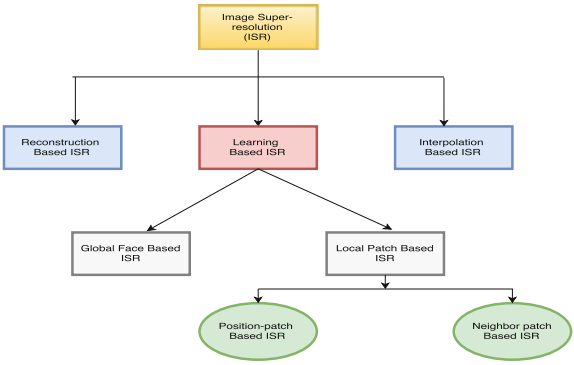


Fig. 1 Classification of image super-resolution mechanisms

On the basis of the type of input image, the classification of existing ISR mechanism can be (i) domain-specific (e.g., text, face) ISR and (ii) generic ISR. The detailed classification of face ISR mechanism by their synthesizing techniques and schemes is given in Fig. 1.

Differential evolutionary algorithm (DE) [18] is one of the popular, nature-inspired meta-heuristic. It has the prime capability to seek the optimal solution in the large search spaces efficiently. Faster convergence and few parameters to adjust are the two major advantages of using DE. In the last decade, there is major growth observed in use of DE. Furthermore, lots of DE variants are also proposed to enhance the efficiency of it by incorporating different mutation, crossover, and selection operators [5, 13, 14]. DE has shown good performance in the field of image processing. Saraswat et al. [17] have employed DE to effectively segment the leukocytes from strained images. Ali et al. [1] presented synergetic DE for multilevel image thresholding of images. A software system based on DE has been proposed by Falco et al. [4] for registering 2D satellite images. Motivated by these wide applications of DE, in this work DE is used as optimization algorithm for minimizing the least square estimation problem to obtain the best weights which are used for generating HQHR image from the LQLR one.

Rest of this work is organized as follows: Sect. 2 gives the brief overview of the work done in the field of ISR. The formulation and methodology of the proposed framework are given in Sect. 3. Simulation and discussion about the results are given in Sect. 4. Finally, the conclusion of the work is given in Sect. 5.

2 Literature Review

In the field of ISR, lot of work have been published in last one and half decade. In this paper, we are mainly concentrating on the domain-specific ISR, i.e., face ISR. The brief overview of some good work done in the direction of face ISR is discussed below.

The first face ISR framework is proposed by the Backer and Kanade [2] in 2000. After this, numerous methods are proposed. Most of the existing methods [6–9, 12, 15] used least square estimation techniques to reconstruct the output HQHR image. Due to the high synthesizing effectiveness of the position-patch-based mechanism over other, nowadays most of the researchers are focusing on these techniques to develop face ISR frameworks or models. In this order, Ma et al. [12] introduced the objective function to minimize least square problem called least square representation (LSR). This objective function gives better reconstruction weights than the previous models. The drawback of this is that it gives biased results when the dimensions of training space are smaller than test dimensions. To overcome this problem, Jiang [7] incorporate the concept of locality in the LSR model called locality constraint representation (LcR). This work further improved in [6, 8, 9, 11] by imposing the various schemes such as sparsity, smooth regression, smooth sparse representation, iterative locality, constraint representation. Recently, some convolution neural network-based ISR models have been introduced to solve the problem of image super-resolution. In this direction, deep neural network-based model is proposed in [10, 21]. The work done in [10, 21] utilized the sparse prior to produce the sharp and HQHR images from the LQLR images. Some other works like [3] have also been utilized deep neural network for the same purpose. The limitation of these models is that they required lot of processing power and time for training.

In this work, a novel attempt is made to find the best reconstruction weights using differential evolutionary (DE) algorithm for generation of HQHR images from the test LQLR image called SRDE framework. Moreover, the proposed SR framework also incorporates sparsity and locality by introducing the new mathematical equations for upper-bound and lower-bound parameters of DE algorithm.

3 The Proposed Face ISR Framework Using DE

Motivated by many position-patch and learning-based[1] ISR frameworks and models [12, 16] in which least square error minimization mechanism is used to estimate the reconstruction weight vector for input image patch from the same position training LQLR patches for synthesizing the resultant HQHR image patch, in this work a novel ISR framework using differential evolutionary (DE) algorithm is proposed.[2] Here, DE algorithm is used to minimize the least square estimation problem to obtain the vector of optimal weights. The mathematical formulation of the least square estimation problem which is used as the fitness function in the DE algorithm can be defined as:

[1] In the learning- and position-patch-based ISR frameworks, each input patch firstly represented as a linear combination of same position training LQLR images patches and then counterpart HQHR patches with obtained weights are used to compute output HR patch.

[2] Note: In this work, we are following the patch-based methods, hence each image is divided into small patches and execute each patch separately.

$$W(a,b) = \underset{W(a,b)}{\arg\min} \left\| I_l^{In}(a,b) - \sum_{k=1}^{K} I_l^{Tr^k}(a,b) W_k(a,b) \right\|_2^2 \quad s.t. \sum_{k=1}^{K}, \tag{1}$$

where (a, b) denotes the position of the patch in the image, I_l^{In} is the test image, I_l^{Tr} is the training LQLR image set in which particular image is denoted as $I_l^{Tr^k}$ and K is the K-nearest images in the training. Here, W is the weight vector, i.e., $W(a, b) = \{W_1(a, b), W_2(a, b), \ldots, W_K(a, b)\}$.

The upper bound (Ub) and lower bound (Lb) for searching W are decided as follows in our proposed method:

$$Ub = \alpha + \left(\frac{1}{1 + d_k}\right) \times \beta, \tag{2}$$

and

$$Lb = (1 - d_k) \times \gamma, \tag{3}$$

where α, β, γ are the tuning parameters and d_k is the square Euclidean distance between input patch and kth training patch.

The procedure for minimizing the function (1) using DE to find the optimal weights is given as follows:

1. Initialize the population of DE randomly in the search space decided by Eqs. (2) and (3).
2. Evaluate the fitness of solutions using function given in Eq. (1).
3. Mutate the solutions using standard DE mutation strategy and store them in *TEMP*.
4. Apply the binomial crossover between parent solutions and mutated solutions of *TEMP* to create an offspring.
5. Select the solution having higher fitness between parents and offspring.
6. Repeat the steps 3 to 5 until the maximum number of iterations.
7. Return the solution which is having best fitness, i.e., vector of optimal weights $W(a, b)$ for patch (a, b).

The obtained weights from above steps are multiplied with the same position patches of training HQHR images to generate output HQHR patch. This can mathematically be represented as below:

$$I_h^{Op}(a,b) = \sum_{k=1}^{K} I_h^{Tr^k}(a,b) W_k(a,b) \tag{4}$$

where $I_h^{Op}(a, b)$ is the output HQHR image patch, I_h^{Tr} is the HQHR image set in which particular image is denoted as $I_h^{Tr^k}$. Finally, the output face is obtained by merging all small HQHR patches. Average the overlapping pixels content.

Fig. 2 Sample FEI images

4 Simulation and Analysis of Outcomes

Here, the detailed description on the experimental outcomes is given to conclude the success of the proposed SRDE framework.[3] The performance of the proposed SRDE framework for reconstruction is compared with Bicubic and EignSR [20] methods. Here, FEI dataset [19] is used for experiments. Two quality metrics, namely PSNR and SSIM [22], are utilized to judge the rebuilding capability of the proposed and existing works.

4.1 Database Description and Parameter Setting

In this work, out of total 400 images of FEI image set, 380 images are used for training and 20 images for testing in such a way that test images are completely absent from the training set. Here, the size of ground truth (GT) HQHR FEI images is set to 120 $\times$ 100 pixels. To make LQLR images, we reduce the scale of GT images by a factor 4 and adding the blur (4 $\times$ 4 average filters). The values of tuning parameters α, β, and γ are experimentally set to 0.09, 1, and 0.001, respectively. Sample GT images are shown in Fig. 2. Apart from above parameters, the DE algorithm also used four self-parameters, i.e., population size,[4] a number of iterations, crossover rate, and F. In this work, the values set for these parameters are as 150 (K), 500, 0.9, and 0.5, respectively.

4.2 Results on FEI Dataset Faces

To confirm the information rendering capability of the proposed SRDE framework, simulation is performed on twenty test human faces from FEI. Simulation results are given in Table 1. Improvement of SRDE over Bicubic and EignSR [20] can also be analyzed from bar chart given in Fig. 3.

[3]For code, mail to: ershyamrajput@gmail.com.

[4]The K-nearest patches for each input patch from the LQLR training image patches are selected at runtime before calling the DE algorithm for optimization for each input patch.

Table 1 The average PSNR and SSIM for twenty test images

Methods	Performance Metrics	
	PSNR (dB)	SSIM
Bicubic	27.71	0.8406
EignSR [20]	28.01	0.7502
SRDE	28.76	0.8591

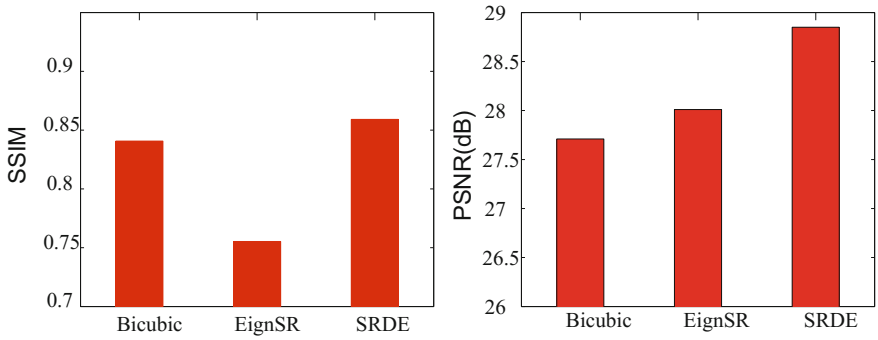


Fig. 3 The average SSIM and PSNR of the Bicubic, EignSR [20], and SRDE for 20 test images

To verify the effectiveness of the differential evolutionary algorithm in the field of image reconstruction, results for twelve out of twenty randomly selected test images are given in Fig. 4. As evident from the figure that the SRDE approach achieved significant gain over the Bicubic interpolation for both metrics (PSNR and SSIM).

Since this paper is targeting the most sensitive object which has various applications (e.g., surveillance, human recognition, identification) in real world, the visual verification of the resultant images is also important. For this purpose, visual results for Bicubic, EignSR [20], and SRDE for seven FEI subjects are shown in Fig. 5.

4.3 Discussion

Table 1 shows the comparison of the proposed SRDE framework with Bicubic and EignSR [20] regarding their PSNR and SSIM results. The comparison can more clearly be analyzed from the bar chart shown in Fig. 3. These results show that the proposed framework gains the better performance over Bicubic and EignSR [20] methods. The improvement of SRDE over the Bicubic method is higher both for PSNR and SSIM. It is typical because of the Bicubic method smooth the content of the input image, lead to missing edge and structure information of the face images. The improvement of SRDE regarding SSIM than PSNR over EignSR [20] is much better. This is because that EignSR [20] depends on a principal component analysis (PCA)-based statistics, which cannot manifest the data distribution whenever the

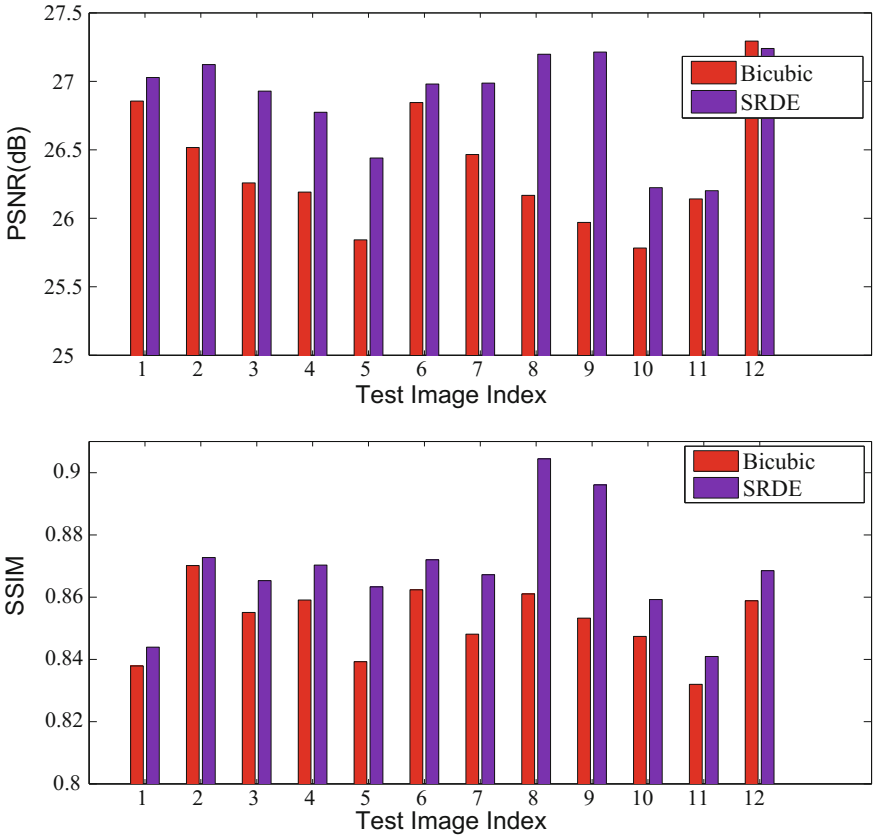
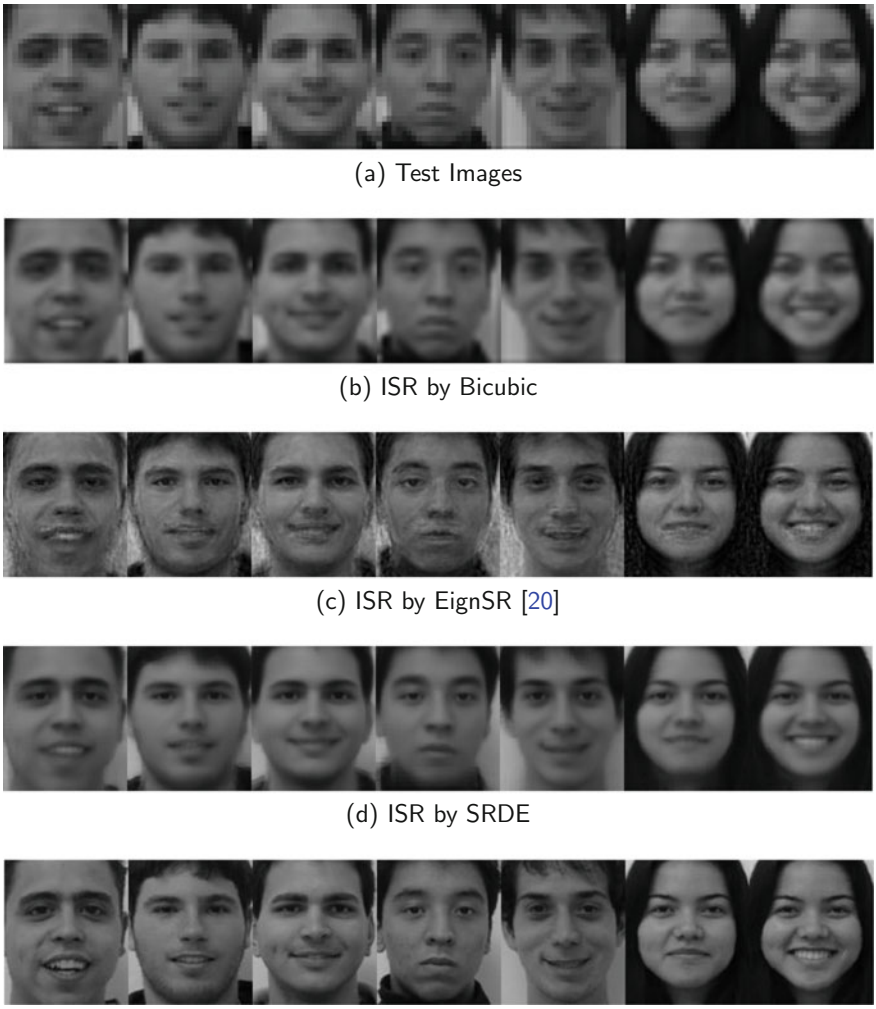


Fig. 4 The performance gain of SRDE over Bicubic for twelve randomly selected test faces

size of training space (i.e., 360) is not greater than the data dimension (test LQLR image size, i.e., $750 = 30 \times 25$ pixels).

The effectiveness of our framework can also be verified from Fig. 5. Hallucinated images of Bicubic method completely lose the high-structural and edge information which is essential for the face recognition that can be visible from second row of Fig. 5. The resultant faces of EignSR also have shriveled and wrinkled effects that may be seen from the third row of Fig. 5. Hallucinated faces of our SRDE framework have more detailed discriminative information than the other, and these faces are almost similar to the ground truth images (verify from the fourth row of Fig. 5).



(a) Test Images

(b) ISR by Bicubic

(c) ISR by EignSR [20]

(d) ISR by SRDE

(e) Ground Truth Images

Fig. 5 Performance of different methods regarding face visual quality

5 Conclusions with Future Directions

Due to the highly structured shape of the human face, facial discriminative information preservation during image super-resolution is challenging. To overcome this issue in this paper, a novel attempt to incorporating differential evolutionary (DE) algorithm for generating hallucinated face from the test face is done. Here, DE is utilized for optimizing the least square estimation problem to generate the vector of optimal weights. Moreover, a new formulation for upper-bound and lower-bound

parameters of DE is also introduced to obtain the important constraint of image super-resolution, i.e., sparsity and locality. The simulation results show that proposed SRDE framework outperforms the Bicubic and EignSR models.

As we have discussed, locality and sparsity are important in face super-resolution. To achieve these two properties, in this work we have introduced upper-bound and lower-bound equations along with the fitness function. The future extension of this work could be the designing of more appropriate upper-bound and lower-bound equations to achieve better reconstruction.

References

1. Ali, M., Ahn, C.W., Pant, M.: Multi-level image thresholding by synergetic differential evolution. Appl. Soft Comput. **17**, 1–11 (2014)
2. Baker, S., Kanade, T.: Hallucinating faces. In: Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), pp. 83–88 (2000)
3. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intelligen. **38**(2), 295–307 (2016)
4. Falco, I.D., Cioppa, A.D., Maisto, D., Tarantino, E.: Differential evolution as a viable tool for satellite image registration. Appl. Soft Comput. **8**(4), 1453–1462 (2008)
5. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans. Syst. Man Cybernet. Part B (Cybernetics) **42**(2) 482–500 (2012)
6. Jiang, J., Chen, C., Ma, J., Wang, Z., Wang, Z., Hu, R.: Srlsp: a face image super-resolution algorithm using smooth regression with local structure prior. IEEE Trans. Multimed. **19**(1), 27–40 (2017)
7. Jiang, J., Hu, R., Wang, Z., Han, Z.: Noise robust face hallucination via locality-constrained representation. IEEE Trans. Multimed. **16**(5), 1268–1281 (2014)
8. Jiang, J., Ma, J., Chen, C., Jiang, X., Wang, Z.: Noise robust face image super-resolution through smooth sparse representation. IEEE Trans. Cybernet. **PP**(99) 1–12 (2016)
9. Jiang, J., Chen, C., Huang, K., Cai, Z., Hu, R.: Noise robust position-patch based face super-resolution via tikhonov regularized neighbor representation. Informat. Sci. **367–368**, 354–372 (2016)
10. Liu, D., Wang, Z., Wen, B., Yang, J., Han, W., Huang, T.S.: Robust single image super-resolution via deep networks with sparse prior. IEEE Trans. Image Process. **25**(7), 3194–3207 (2016)
11. Liu, L., Chen, C.L.P., Li, S., Tang, Y.Y., Chen, L.: Robust face hallucination via locality-constrained bi-layer representation. IEEE Trans. Cybernet. **PP**(99) 1–13 (2017)
12. Ma, X., Zhang, J., Qi, C.: Hallucinating face by position-patch. Pattern Recogn. **43**(6), 2224–2236 (2010)
13. Qian, W., li, A.: Adaptive differential evolution algorithm for multiobjective optimization problems. Appl. Math. Computat. **201**(1) 431–440 (2008)
14. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791 (2005)
15. Rajput, S.S., Arya, K.V., Singh, V.: Robust face super-resolution via iterative sparsity and locality-constrained representation. Inform. Sciences (2018). https://doi.org/10.1016/j.ins.2018.06.050
16. Rajput, S.S., Singh, A., Arya, K.V., Jiang, J.: Noise robust face hallucination algorithm using local content prior based error shrunk nearest neighbors representation. Signal Process. **147**, 233–246 (2018)
17. Saraswat, M., Arya, K., Sharma, H.: Leukocyte segmentation in tissue images using differential evolution algorithm. Swarm Evolution. Computat. **11**, 46–54 (2013)

18. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optimizat. **11**(4), 341–359 (1997)
19. Thomaz, C.E., Giraldi, G.A.: A new ranking method for principal components analysis and its application to face image analysis. Image Vis. Comput. **28**(6), 902–913 (2010)
20. Wang, X., Tang, X.: Hallucinating face by Eigen transformation. IEEE Trans. Syst. Man Cybernet. Part C (Appl. Rev.) **35**(3) 425–434 (2005)
21. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 370–378 (2005)
22. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)

Noise-Robust Low-Resolution Face Recognition Using SIFT Features

K. V. Arya, Shyam Singh Rajput and Shambhavi Upadhyay

Abstract In many everyday life applications, the captured images of blurry, noisy, and small in size can be called as low-resolution images which make the task of face recognition challenging. To minimize this problem, a new scale-invariant feature transformation (SIFT) descriptors-based noise-robust low-resolution face recognition model is developed in this work. Due to the robustness of the SIFT descriptors against blur, in this work SIFT features are utilized for the purpose of feature extraction and matching. The results obtained from the experiments performed on two public JAFFE and ORL face databases show that the proposed face recognition model works better than the existing algorithms for low-resolution face images.

Keywords SIFT features · Face recognition · Low-resolution

1 Introduction

For more than two decades, face recognition (FR) has turned out to be an important working area in research, leading to the development of many encouraging practical applications of face recognition systems [1]. This is common that existing FR systems achieve good performance under supervised conditions, e.g., adequate information can be detected from faces, but these systems face a problem with low-resolution (LR)

K. V. Arya
Department of Computer Science & Engineering, Institute of Engineering & Technology, Lucknow 226021, India
e-mail: kvarya@ietlucknow.ac.in; kvarya@gmail.com

S. S. Rajput (✉)
Multimedia and Information Security Research Group, ABV—Indian Institute of Information Technology & Management, Gwalior 474015, India
e-mail: ershyamrajput@gmail.com

S. Upadhyay
Manipal University, Manipal 576104, Karnataka, India
e-mail: shambhaviu02@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_49

images [2] which are blurry and noisy. Such LR problems are faced in many surveillance cameras where less facial features are detected because the person is far away. In such cases, captured image is of low-resolution (LR) and absent in comprehensive features which are necessary for face recognition. Various characteristics of faces recorded by surveillance sensors/cameras are inadequate resolution due to variation in noise, atmospheric blur, variation in illumination, and pose that negatively affect the performance of face recognition (FR) systems. Traditionally, a primary focus in the field of FR has been on identifying faces over the change in pose and uneven illumination, but interest in dealing with low-resolution recognition due to noise has grown immensely. In this work, we are concerned about the obtaining of essential characteristics from noisy LR face images and match that with the corresponding high-resolution (HR) gallery of images.

Rest portion of this work is represented as follows: Sect. 2 gives the brief overview of the existing LR face recognition methods. The working of our method is explained in Sect. 3. Section 4 gives the simulation and analysis about the outcomes. Section 5 gives summary of this work.

2 Related Work

Several face recognition (FR) methods [6–14] have been reported in the literature that can handle various factors like pose, illumination efficiently. The matching between different dimensions images (test LR image and gallery HR images) cannot be performed directly. There are three ways to handle this problem [14]: (i) Downscale the entire gallery images and match LR images in LR space; (ii) high-resolution version of LR image can be obtained using super-resolution [3] and then match; (iii) difference in the appearance of images in different resolutions can be reduced, and traditional recognition algorithms can be applied in the unified space.

Recently, numerous super-resolution frameworks have been introduced which involve learning to solve LR problems. Baker and Kanade [4] proposed "*face hallucination*" using Bayesian formulation to get the HR version from an input LR face image. In this, it gets the important information from parent structure with the help of training images. Lui et al. [5] proposed an algorithm which is a two-step process for face image super-resolution that included both global parametric principal component analysis (PCA) and a local nonparametric Markov random field (MRF). These methods focus on hallucination face image. Drawback of probabilistic model-based face hallucination was that it needs large number of training images. Face hallucination created interest after the introduction of these methods, and a vast number of different models have been introduced that focus on face hallucination. Wang and Tang [6] used the same idea of face hallucination method through the eigen-transformation method. In this method, PCA has been used to represent LR image as a linear combination of HR training images, but this method cannot maintain overall smoothness, especially at locations around face and mouth. Chakrabarti et al. [7] further improvised the idea of face hallucination by incorporating kernel

PCA which is used for image magnification by projecting an interpolated LR image version on the HR image subspace. Park and Savvides [8] developed an algorithm which deals with all the different images equally. This algorithm uses PCA to break down a test LR image into several prototypic faces and uses recursive error back-propagation method to get HR image back. Models stated above are complete and fail to provide efficiently the accurate single specifics. To overcome this drawback, many techniques have been introduced that divide an image into many small patches with an assumption that same local geometric structure is shared between HR and LR image patches. With this assumption, Chang et al. [9] used locally linear embedding (LLE) [10], concluding that training LR images and corresponding HR once form manifold with same local geometry. By taking inspiration from Change's method, Ma et al. [11] introduced a patch-centered position-dominant hallucination technique where position information is used in addition to image feature to recreate a novel image. This method concluded that information about position is essential for analysis of the face. The existing methods are unbeatable in their domain, but these methods are time-consuming and hence not right for existing world FR applications. Therefore, above-proposed learning-based methods give unsuitable results for LR face problems.

To minimize the disadvantages of previous systems, methods based on common space have been proposed which focus on manifold learning [12–14]. Li et al. [12] suggested a coupled locality preserving mapping (CLPM) technique that projects different resolution images, i.e., HR images and LR images into a unified common feature space which makes the job of classification easy. It is assumed that images of same subject belong to the same manifold and those belonging to different subject belong to different manifold. However, CLPM uses unsupervised method (without label information) to identify correct face image which may leave discriminative information, and this can lead to misclassification. To over this drawback, Wilman and Yuen [13] developed a discriminative SR method (DSR) that includes the class label information which was absent in CLPM method. This method attains discriminability through label information. However, DSR method calculates within-class and between-class distance, considering all the image samples to be equal. This represents the global geometric structure of manifold and drops the local structure, which is considered effective for face recognition. To handle the drawbacks mentioned above, Jiang et al. [14] proposed an algorithm called coupled discriminant multi-manifold analysis (CDMMA). It considers simultaneous learning from local geometric structure and neighbor information such that the intra-manifold distance is maximized and inter-manifold distance is reduced, and thus, better discriminative features can be extracted to give better face recognition results.

The above methods calculate Euclidean distance [15] between HR image pixels and LR image pixels. They give dissatisfactory recognition rate when tested on blurred low-resolution images with Gaussian noise [16] because they cannot effectively calculate distance between HR image pixels and LR noisy image pixels. To overcome this drawback, we proposed a noise-robust feature extraction method using SIFT. It extracts important and prominent features from images in the form of

descriptors. It effectively calculates the Euclidean distance between noisy LR image pixels and HR image pixels because SIFT is flexible to noise in images and handles it efficiently.

3 The Proposed Method

Recently, the increasing importance of using local features like SIFT for matching facial images has been noticed. To handle the drawback mentioned above, the proposed SIFT feature extraction is used to extract local features from images. Local feature descriptors are used to give information about the image pixel from its content in local region and hence proved that local features are important in FR in unrestricted surroundings with variation in pose, lighting. Local descriptors directly compare images with different resolutions with required changes in the scale while computing descriptors, getting an advantage over holistic face representations. SIFT image features give features of an image that are not affected by factors such as object scaling and rotation. One major advantage SIFT features have over other existing methods is that it is resilient to the effects of "noise" in the image. Therefore, our proposed SIFT method can be used to extract local feature descriptors from noisy images. A four-stage filtering approach is applied to extract local features using SIFT algorithm:

Scale-Space Extrema Detection: In the first step, Laplacian of Gaussian [17] (LOG), which acts as a blob detector, is computed for different variances of Gaussian noise denoted by "σ". It helps in detecting blobs due to change in σ. SIFT utilizes difference of Gaussian (DoG), an approximation of LoG due to costly nature of LoG. The DoG is taken as the variation of Gaussian blurring of an image with two distinct σ values. This arrangement is made for various octaves of the image in Gaussian pyramid. Once the DoG is detected, images are explored for regional extrema over scale and space. It is a potential keypoint if it is a local extremum.

Keypoint Localization: To get better results, potential keypoints found earlier are refined. Taylor series expansion is used to refine the keypoints and get better location of extrema. The extrema obtained are selected on the basis of a threshold value set for the intensity. The obtained extrema are denied if its intensity is less than the fixed threshold. This leads to the selection of strong keypoints only by eliminating any edge and low-contrast keypoints.

Orientation Assignment: Every obtained keypoint has specified an orientation to achieve image rotation invariance. The calculation of gradient magnitude and direction is completed by considering a neighborhood nearby the keypoint location by scale. The highest peak along with any peak above 80% of orientation histogram created with 36 bins is used to determine the orientation. The keypoints with identical scale and position but inconsistent direction are created which helps in increasing stability of matching.

Keypoint Descriptors: After taking a 16×16 region around the keypoint, the neighborhood is further separated into 16 sub-blocks of 4×4 size. For each sub-

block, 8-bin orientation histogram is formed. Hence, we get a descriptor of 128 bin values. These keypoint descriptors represent features of the fiducial keypoint.

Though these feature descriptors obtained are known to give good recognition rate with changes in pose and scale, using them for matching LR test image with change in noise to a HR image gallery has not been done yet. To study the effectiveness of SIFT features across variations of noise and other outside imaging aspects, we perform a recognition experiment on the Japanese Female Facial Expressions [18] (JAFFE) and ORL [19] database with HR frontal gallery and LR test images with variation in poses. We use SIFT descriptors at particular points as the features for performing recognition.

The working of the proposed noise-robust face recognition method can be seen from the flowchart given in Fig. 1. LR probe images are obtained from their respective HR gallery image by down-sampling HR image, by a factor of 4. Down-sampled image is then induced with Gaussian noise (mean = 0, variance = 0.01) and blurred using average filter, a 2×2 pixel filter. Since the proposed SIFT method requires both LR image and HR image to be in the same dimension, LR test image is resized to HR dimensions (72×72 pixels in case of ORL database). This does not reduce the prominent feature descriptors in LR images. Feature extraction is done using the proposed noise-robust SIFT method. It gives SIFT descriptors at fiducial locations for LR test image (resized) and HR gallery images. The SIFT descriptors from all prominent points are collected together into a single descriptor. Euclidean distance is computed between feature descriptors of probe image and gallery images. After sorting the distance matrix, index matrix is achieved which is used to get the class label of training images. Class label matrix gives the class of training images that have correctly matched with the test image, based on Euclidean distance. Further comparison between class label matrix and test label gives the accuracy of the method.

3.1 Matching

Matching of LR test image keypoint feature descriptors to corresponding HR training images keypoint descriptors is done by calculating nearest neighbors using Euclidean distance. However in some cases, it is possible that second-closest match may be very near to first match due to variation in noise. To deal with this problem, ratio of first-closest distance to second-closest distance is taken in the proposed method. If it is greater than the predefined threshold, then they are rejected. This helps eliminate a good percentage of false matches while discarding only a small percentage of correct matches.

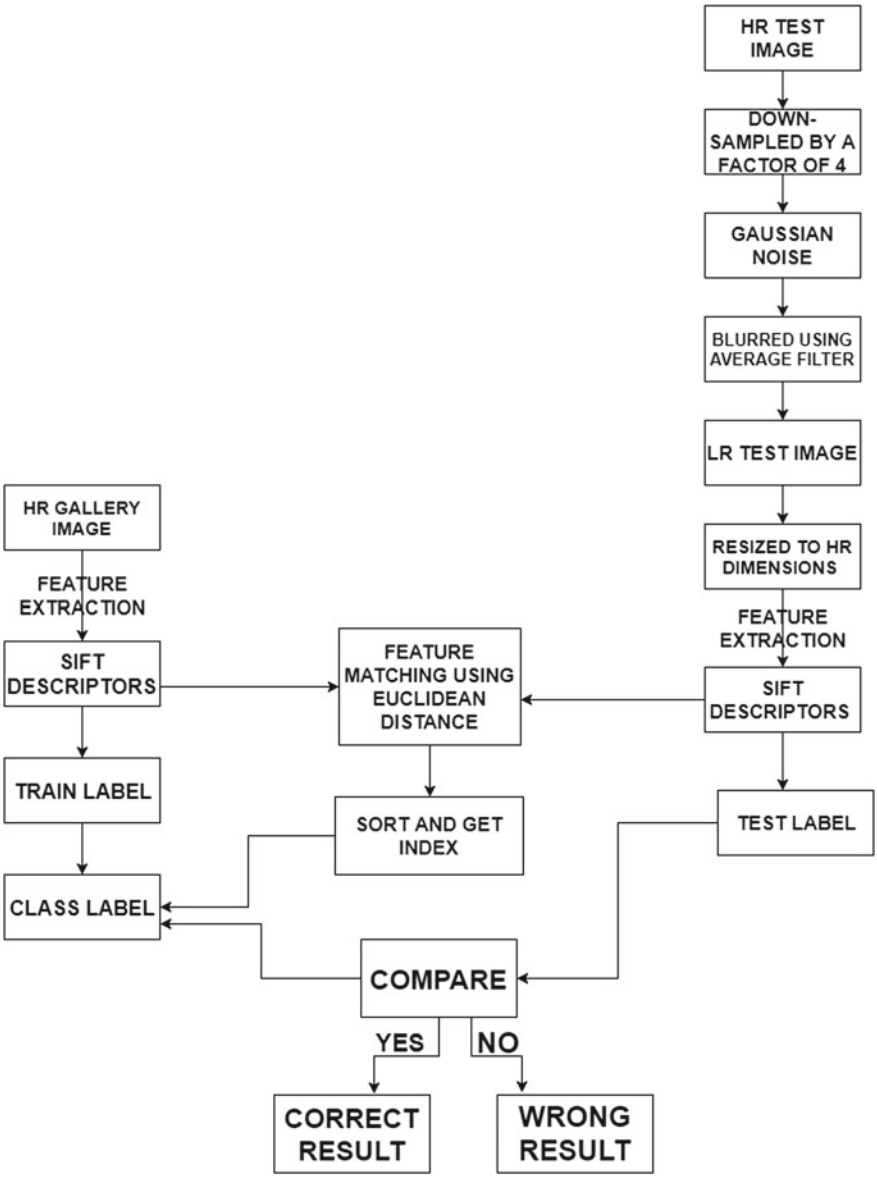


Fig. 1 Working model of the face recognition method

4 Experiments and Results

To evaluate the performance of all the proposed methods, experiments are performed on two popular datasets (JAFFE and ORL). In SIFT, the dimensions of training and test images must be same. Therefore, in all the experiments test LR faces are preprocessed using bicubic interpolation for resigning them to the size of the gallery HR faces. Another preprocessing is all noisy images are smoothed through the average filter. We apply this second preprocessing with all existing methods for fair comparison. The threshold parameter value is experimentally set to 0.6.

4.1 Description of Face Datasets

All the experiments given in this paper are achieved on Japanese Female Facial Recognition (JAFFE) and ORL database. In this paper, all the face recognition systems are using executed on a Windows 8 with Pentium-IV machine having 3.20 GHz processor and 512 MB RAM.

The JAFFE facial image set consists of 213 faces with 7 different expressions. There are 10 Japanese female subjects. In these experiments, frontal pose of all the subjects is considered and all the images are under same lighting, giving 190 images in total. Training set contains 120 h images with size 72 × 72 pixels, 12 images for each subject. Testing set contains 70 LR images with size 18 × 18 pixels, with 256 gray levels per pixel. Figure 2a shows sample JAFFE dataset.

The ORL facial dataset contains 400 images of size 112 × 92. There are 40 subjects, 10 images per each person. The images are under same lighting with different facial expressions. In all experiments, different poses under same illumination are chosen. Training set contains 240 h images with size 72 × 72 pixels, 6 images for each subject. Testing set contains 160 LR images with size 18 × 18, with 256 gray levels per pixel. Figure 2b shows sample ORL database.

4.2 The Outcomes Comparison

The proposed noise-robust face recognition method has been compared with four popular existing methods. The values in Table 1 show the recognition rate on ORL and JAFFE database when rank is 1.

In Fig. 3, the accuracy rate for different methods on ORL and JAFFE database is given. The proposed noise-robust SIFT feature extraction method achieves the recognition rate of 88.85% and 91.2% on JAFFE and ORL database, respectively. The recognition rates for other methods are less than our proposed method because these methods use Euclidean distance to calculate distance between HR and LR images. Since the input image in our experiments is noisy and blur, existing methods

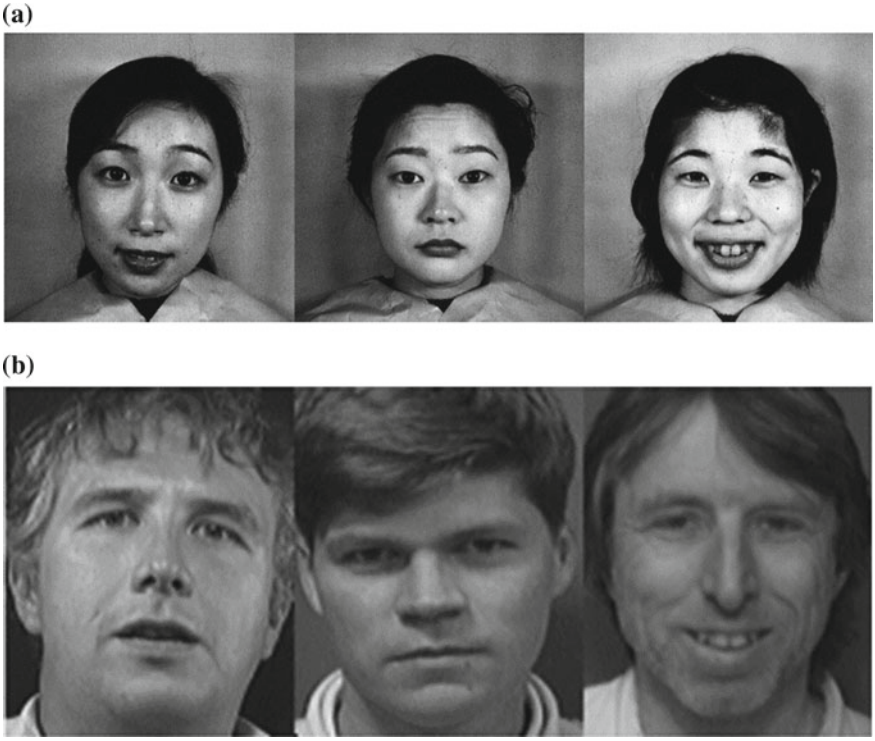


Fig. 2 Sample faces **a** from JAFFE dataset and, **b** from ORL dataset

Table 1 Rank-1 recognition rates of different algorithms on JAFFE and ORL facial images set

Models	Recognition accuracy (%)	
	JAFFE database	ORL database
KNN	22.29	57.38
CM	69.3	77.75
CLPM	77.10	85.75
CDMMA	80.35	88.9
Our method	**88.85**	**91.2**

are not well structured to calculate correct distance between HR, noisy, and blur LR test image. In Fig. 4, cumulative match curve (CMC) for different methods on JAFFE and ORL database, respectively, is given. CMC curve shows variation in performance rate across different ranks for different methods on JAFFE and ORL database. The performance of the existing methods degrades significantly in the presence of noise because in the presence of the noise matching between the features is erroneous. It can be seen from the curve that recognition accuracy increases as rank increases. This is because as rank increases, chances of matching of the test image to correct

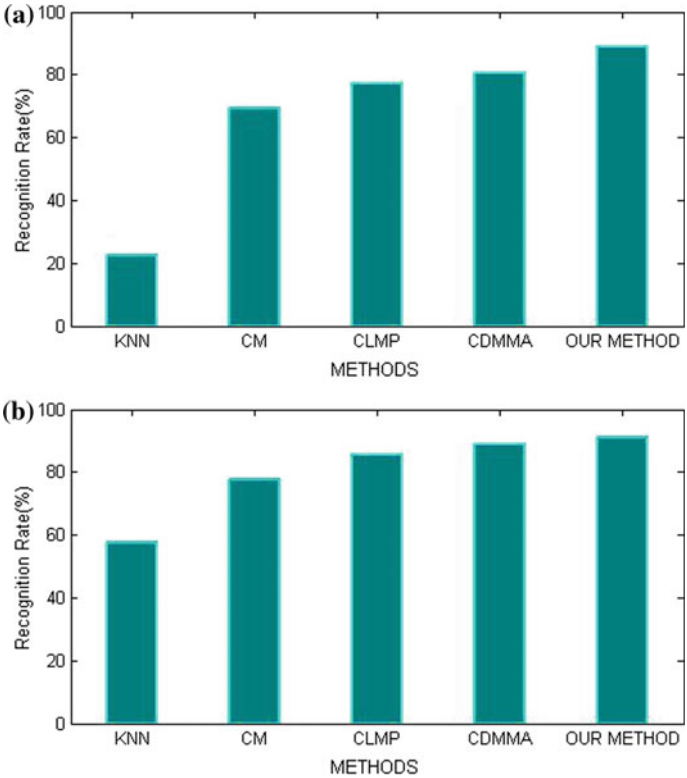


Fig. 3 Results in terms of face recognition rate on **a** JAFFE dataset and, **b** ORL database

training image increase. It can be concluded from the curves that our proposed method achieves recognition rate that is higher than the existing method.

5 Conclusions and Future Direction

In this work, scale-invariant feature transformation (SIFT)-based innovative noise-robust face recognition method has been suggested to answer the problem of face recognition for noisy, blurry, and LR images. The blur-invariant characteristic of SIFT descriptors allows the proposed method to handle the blur and noise in the test images. Experimental outcomes on the JAFFE and ORL facial images datasets show the superiority of the proposed method over other existing methods.

In future, a two-stape model using super-resolution technique [20] could also be developed to provide better recognition accuracy for noisy low-resolution facial images.

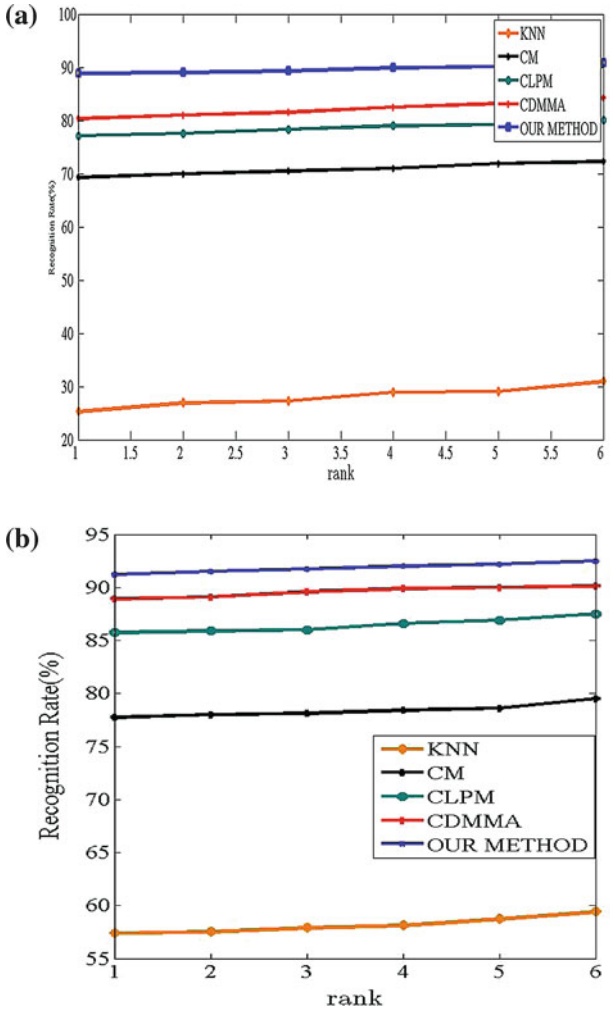


Fig. 4 CMC curve results on **a** JAFFE, **b** ORL database

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: a literature survey. ACM Comput. Surv. **35**(4), 399–458 (2003)
2. Wang, Z., Yang, W., Ben, X.: Low-resolution degradation face recognition over long distance based on cca. Neural Comput. Appl. **26**(7), 1–8 (2015)
3. Rajput, S.S., Singh, A., Arya, K.V., Jiang, J.: Noise robust face hallucination algorithm using local content prior based error shrunk nearest neighbors representation. Signal Process. **147**, 233–246 (2018)
4. Baker, S., Kanade, T.: Hallucinating faces. In: Proceedings of IEEE Conference on Automatic Face and Gesture (FG), pp. 83–88 (2000)

5. Liu, C., Shum, H.Y., Zhang, C.S.: A two-step approach to hallucinating faces: global parametric model and local nonparametric model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. I-192–I-198 (2001)
6. Wang, X., Tang, X.: Hallucinating face by eigen transformation. IEEE Trans. Syst. Man Cybern. Part C **35**(3), 425–434 (2005)
7. Chakrabarti, A., Rajagopalan, A., Chellappa, R.: Super-resolution of face images using kernel pca-based prior. IEEE Trans. Multimed. **9**(4), 888–892 (2007)
8. Park, S.W., Savvides, M.: Breaking the limitation of manifold analysis for super-resolution of facial images. In: Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. I-573–I-576 (2007)
9. Chang, H., Yeung, D., Xiong, Y.: Super-resolution through neighbor embedding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 275–282 (2004)
10. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
11. Ma, X., Zhang, J., Qi, C.: Hallucinating face by position-patch. Pattern Recogn. **43**(6), 2224–2236 (2010)
12. Li, B., Chang, H., Shan, S., Chen, X.: Low-resolution face recognition via coupled locality preserving mappings. IEEE Signal Process. Lett. **17**(1), 20–23 (2010)
13. Wilman, W.Z., Yuen, P.C.: Very low-resolution face recognition problem. IEEE Trans. Image Process. **21**(1), 327–340 (2012)
14. Jiang, J., Hu, R., Wang, Z., Han, Z.: CDMMA: coupled discriminant multi-manifold analysis for matching low-resolution face images. Signal Process. **124**, 162–172 (2016)
15. Saito, T., Toriwaki, J.: New algorithms for euclidean distance transformation of an *n*-dimensional digitized picture with applications. Pattern Recogn. **27**(11), 1551–1565 (1994)
16. Luisier, T. Blu, M.: Unser, image denoising in mixed poisson–gaussian noise. IEEE Trans. Image Process. **20**(3) (2011)
17. Sharif, M., Mohsin, S., Javed, M.Y., Ali, M.A.: Single Image Face Recognition Using Laplacian of Gaussian and Discrete Cosine Transforms, Received 31 Aug 2010; Accepted 24 Oct 2010
18. Deng, H., Clausi, D.A.: Gaussian MRF rotation-invariant features for image classification. IEEE Trans. Pattern Anal. Mach. Intell. **26**(7) (2004)
19. Lyons, M.J., Budynek, J., Akamatsu, S.: Automatic classification of single facial images. IEEE Trans. Pattern Anal. Mach. Intell. **21**(12), 1357–1362 (1999)
20. Rajput, S.S., Arya, K.V., Singh, V.: Robust face super-resolution via iterative sparsity and locality-constrained representation. Information Sciences. https://doi.org/10.1016/j.ins.2018.06.050

Low-Dose CT Image Reconstruction Using Complex Diffusion Regularization

Kavkirat Kaur and Shailendra Tiwari

Abstract The computed tomography (CT) is considered as a significant imaging tool for the clinical diagnosis. Due to the low-dose radiation in the CT, the projection data is highly affected by the Gaussian noise. Thus, there is a demand of a framework that can eliminate the noise and provide high-quality images. This paper presents a new statistical image reconstruction algorithm by proposing a suitable regularization method. The proposed framework is the combination of two basic terms, namely data fidelity and regularization. Minimizing the log-likelihood gives data fidelity term, which represents the distribution of noise in low-dose X-ray CT images. Maximum likelihood expectation maximization algorithm is introduced as a data fidelity term. The ill-posedness problem of data fidelity term is overcome with the help of complex diffusion filter. It is introduced as regularization term into the proposed framework that minimizes the noise without blurring edges and preserving the fine structure information into the reconstructed image. The proposed model has been evaluated on both simulated and real standard thorax phantoms. The final results are compared with the various other methods, and it is analyzed that the proposed model has many desirable properties such as better noise robustness, less computational cost, enhanced denoising effect.

Keywords Computed tomography (CT) · Noise reduction · Maximum likelihood expectation maximization (MLEM) algorithm · Complex diffusion (CD) Gaussian noise

K. Kaur (✉)
Thapar Institute of Engineering and Technology (TIET), Patiala 147004, India
e-mail: kkaur_me16@thapar.edu

S. Tiwari
School of Computer Science & Statistics (SCSS),
Trinity College Dublin (TCD), Dublin, Ireland
e-mail: tiwaris@tcd.ie; shailendra@thapar.edu

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_50

1 Introduction

Nowadays, computed tomography (CT) is widely used in medical imaging for the clinical diagnosis of various health problems like cancer, tumors, bladder stones. CT has become one of the most commonly used medical imaging tools for analyzing the internal organs of the human body. Sometimes, a patient has to undergo a CT for multiple times which could risk the life of a patient due to the high-dose radiations. Though these radiations provide high-quality images, these radiations are harmful. They can cause various health problems like cancer, skin reactions, and other genetic disorders. Low radiation dose decreases the accuracy and quality of the image which affects while diagnosing the problem. In medical imaging, minimization of the radiation dose while maintaining the quality of the image is considered as one of the emerging topics for current research. As low as reasonably achievable (ALARA) principle is used in minimization of the radiation dose.

Different methods have been proposed for minimizing the radiation dose likes [1]: (1) minimizing the exposure time and current in the X-ray tube, (2) decreasing the number of projections, (3) improving software approaches, i.e., the reconstruction methods. There are various methods of image reconstruction that are introduced over the last few decades. Basically, they are broadcasted in two main categories [1], i.e., analytical reconstruction and iterative reconstruction. In analytical reconstruction, filtered backpropagation (FBP) [2] method is used due to its computational efficiency and numerical stability. Generally, the poor quality images are constructed by the analytical method as higher noise is induced during data acquisition. On the other hand, iterative reconstruction methods have the capability to handle the noisy data and reconstruct a better quality image. Iterative reconstruction method is computationally more expensive than analytical reconstruction. In iterative reconstruction, multiple steps are performed iteratively to form a better quality image. Iterative reconstruction is further categorized into two types: algebraic iterative reconstruction and statistical iterative reconstruction (SIR) methods. Statistical iterative reconstruction methods provide better results as compared to the classical iterative reconstruction methods. Various statistical iterative reconstruction techniques are median root prior (MRP) [3], maximum likelihood expectation maximization (MLEM) [4], ordered-subset expectation maximization (OSEM) [5], row-action maximum likelihood algorithm (RAMLA) [6], complete-data ordered-subset expectation maximization (COSEM) [7], adaptive statistical iterative reconstruction (ASIR, GE Healthcare) [8], model-based iterative reconstruction (MBIR, Veo) [9], image reconstruction in image space (IRIS) [10], advanced modeled image reconstruction (ADMIRE) [11], and many more. Among all the techniques, MLEM is one of the most commonly used algorithms for the image reconstruction. MLEM provides better smoothing to the images.

All the techniques discussed above belong to the class of inverse problem that is inherent of the ill-posed nature [3]. Hence to overcome the problem of ill-posed nature of SIR, regularization was introduced. Regularization is used with the basic iterative algorithms to provide a priori information for better-reconstructed images. Numerous age preserving priors have been used from the past few decades such

as Gaussian mixture priors [12], Gibbs smoothing [13], anisotropic diffusion (AD) [14], total variation (TV) [15], anisotropic median diffusion (AMD) [16], complex diffusion (CD) [17], non-local means (NLM) [18]. MLEM with AD was proposed [19] by Yang but was unable to reduce the noise and preserve the edges. Median anisotropic diffusion (AMD) is an improved version of AD which provides better smoothing. Later, He and Huang proposed a model MLEM-AMD [20] which is the combination of MLEM and AMD as regularization term. This model faces problem in detecting the edges. Total variation with MLEM was introduced by Chavez-Rivera [21]. Besides a good approach, sometimes edges are considered as the noise and are removed due to which the results of the reconstructed image decrease. Zhang et al. [22] proposed NLM as regularization term with SIR. But this approach is very time consuming and complex.

Therefore, in this paper, a new statistical reconstruction framework for low-dose CT is introduced. In the proposed model, a new complex diffusion regularization term is combined with basic MLEM iterative methods to handle the Gaussian noise and produce an optimal image with low radiation dose. The complex diffusion proposed by Gilboa et al. [17] is used as the regularization term to maximize the likelihood function, remove the noise, and stop creating false edges.

This paper is organized in various sections. Sect. 2 includes the description of the new proposed model MLEM with CD as regularization term. Result and discussions is discussed in Sect. 3. Conclusion of the paper is included in Sect. 4.

2 Method and Models

The main problem of low-dose CT image reconstruction considered in this paper is an ill-posed problem either due to the scarcity of X-ray projections data or due to the presence of noise and other inconsistencies in the acquired sinogram CT data [21]. Therefore, the image estimation that directly optimizes the maximum likelihood (ML) criterion can be very noisy and unstable. So, researchers reformulate this problem with the maximum a posteriori (MAP) estimation by applying a prior term to penalize the solution. The prior term enables us to incorporate available information or expected properties of the image to be reconstructed. Perona and Malik (P-M) have introduced the first partial differential equation (PDE) based on nonlinear anisotropic diffusion (AD) filter [14]. Although this method can retain the edges to some amount, it cannot retain the edge details effectively for very noisy images. It also causes the blocking staircase effect in the final outcome. In addition to that, it leads to smoothening out only the strong magnitude edges that affect over smoothing and lost fine edges. Therefore, to preserve the minor information as well as to produce the quality images, more robust regularization is needed. Here, in this paper, a new nonlinear complex diffusion (CD) prior is introduced as a regularization term in CT image reconstruction to reduce the limitations of AD method. The CD prior consists of two terms real part and an imaginary part which behaves as a real diffusion process and a second-order derivative of the real part, respectively.

Here, we incorporate nonlinear complex diffusion (CD) filter as a regularization term with SIR method for low-dose image reconstruction. The reconstructed image can be achieved by minimizing the cost function. In variational framework [22, 23], the minimization problem can be read as:

$$f^* = \arg\min_{f \geq 0} E(f) \tag{1}$$

where f^* is the estimated solution, i.e., the reconstructed image obtained by minimizing the energy functional E(f); E(f) is given by:

$$E(f) = E_1(f) + \lambda\, E_2(f) \tag{2}$$

In Eq. (2), $E_1(f)$ is called as a data fidelity term that models the statistic of measured projection data, whereas the second term represents the regularization or smoothing term which is used as a priori knowledge of the image to be reconstructed. The balancing parameter λ is used to control the trade-off between data fidelity and regularization.

In MAP criteria, minimizing the log-likelihood function gives the data fidelity term which portrays the distribution of noise in low-dose X-ray CT images [1]. Therefore, in the noisy environment, the normal (Gaussian) probability distribution function (PDF) of the measured data is defined as:

$$P(g|f) = \prod_{i=1}^{M} P(g_i|f_i) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(g_i - f_i)^2}{2\sigma_{y_i}^2}\right) \tag{3}$$

where $f = (f_1, f_2, \ldots, f_N)^T$ represents the true image vector to reconstruct the image, N is the number of voxels, $g = (g_1, g_2, \ldots, g_M)^T$ is the measured data, and M *is* the total number of sampling points in the projection data. By taking the negative log-likelihood of the approximated data after elimination of constant and irrelevant terms, results in the function are given by:

$$E_1(f) = \ln\, P(g|f) = \sum_{i=1}^{M} \left\{ \frac{(g_i - f_i)^2}{2\sigma_{y_i}^2} \right\} \tag{4}$$

In the MAP estimation, the regularization is used as priori information aiming to provide the solution of the ill-posedness problem in the reconstruction algorithm [12]. Since the ideal requirement for any regularization term is to achieve the optimal reconstructed image with preserved features while keeping low radiations, a robust regularization method is required. Therefore, we choose a nonlinear complex diffusion (CD) regularization term due to the following reason: (a) It preserves the edges and minor information in the noisy projection data, and (b) it provides reconstructed image in an acceptable computational time. Thus, the regularization function can be written as:

$$E_2(f) = \arg\min(div(c(\mathrm{Im}(f))\nabla f)) \tag{5}$$

where Im(.) is the imaginary value and the diffusion coefficient $c(\mathrm{Im}(f))$ is defined as:

$$c(\mathrm{Im}(f)) = \exp(i\theta)/1 + \left(\frac{\mathrm{Im}(f)}{k\theta}\right)^2 \tag{6}$$

where k is defined as threshold parameter. The value of phase angle θ must be less than one, i.e., ($\theta \ll 1$). The results of the proposed framework can be acquired by substituting Eqs. (4) and (5) into Eq. (2), to get the modified energy function written as:

$$E(f) = \arg\min_{f \geq 0} E(f) = \sum_{i=1}^{M} \left\{ \frac{(g_i - f_i)^2}{2\sigma_{y_i}^2} \right\} + \lambda \int_{\Omega} (div(c(\mathrm{Im}(f))\nabla f))d\Omega \tag{7}$$

The functional $E(f)$ is defined on the set of $f \in BV(\Omega)$ such that $\log f \in L^1(\Omega)$ and f must be positive everywhere.

$$f^* = \sum_{i=1}^{M} \left\{ \frac{(g_i - f_i)^2}{2\sigma_{y_i}^2} \right\} + \lambda div(c(\mathrm{Im}(f))\nabla f), \quad with \quad \frac{\partial f}{\partial \overrightarrow{n}} = 0 \; on \; \partial\Omega \tag{8}$$

The consideration of Euler–Lagrange minimization technique combined with gradient descent approach provides the optimum solution which is given by:

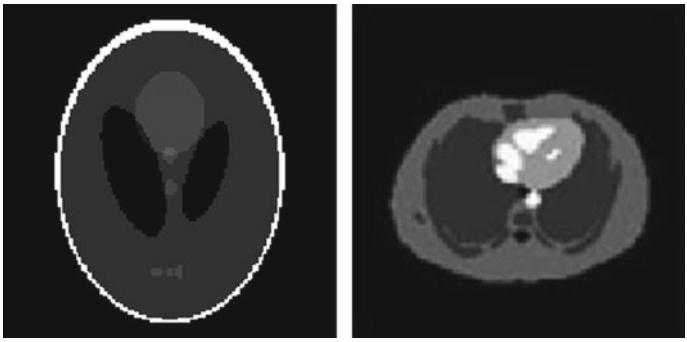
$$f^* = \frac{(g_i - f_i)}{\sigma_i^2} + \lambda div(c(\mathrm{Im}(f))\nabla f), \quad with \quad \frac{\partial f}{\partial \overrightarrow{n}} = 0 \; on \; \partial\Omega \tag{9}$$

For digital implementations, Eq. (9) can be discretized using finite differences schemes [23]. After discretization, the proposed model reads as:

$$\frac{\widehat{f}_j^{t+1} - \widehat{f}_j^{t}}{\Delta t} = \frac{(g_j - f)}{\sigma_j^2} + div\left(c\left(\mathrm{Im}\left(\hat{f}_j^t\right)\right)\nabla \hat{f}_j^t\right), \quad with \quad \frac{\partial x}{\partial \overrightarrow{n}} = 0 \; on \; \partial\Omega \tag{10}$$

$$\widehat{f}_j^{t+1} = \widehat{f}_j^{t} + \Delta t\left[\frac{(g_i - f_i)}{\sigma_i^2} + div\left(c\left(\mathrm{Im}\left(\widehat{f}_j^t\right)\right)\nabla \widehat{f}_j^t\right)\right] \tag{11}$$

To stabilize the discretized version solution of Eq. (11), the von Neumann analysis [23] method is used that requires setting the value of $\Delta t/(\Delta f)^{\mathrm{t}} < 1/4$. The grid size is set to $\Delta f = 1$; then, $\Delta t \leq 1/4$. Therefore, the Δt value is set to 1/4 for stability of Eq. (11).



(i) (ii)

Fig. 1 Test phantoms considered in the experimental setup as simulated modified Shepp-Logan head phantom and real thorax phantom

3 Result and Discussions

In this section, the comparison has been made between the reconstruction algorithm of the proposed MLEM + CD approach and existing standard approaches available in the literature, namely MLEM [4], MLEM + AD [19], MLEM + TV [21], MLEM +NLM [22], and MLEM + AwTV [24] for X-ray CT with low-dose imaging. The proposed and the compared methods have been implemented in MATLAB v2015b on a PC with Intel(R) No. of Core/No. of threads (4/8) i7 CPU 630 @ 3.6 GHz, 8.00 GB RAM, and Microsoft Windows(64-bit) OS.

To validate the performance of the proposed approach, both simulated and real test cases phantoms have been considered in this paper as shown in Fig. 1. The description of both the test phantom (such as the design parameters for the simulated test phantoms) has been presented separately as follows.

In simulated test case, the modified Shepp-Logan head phantom (shown in Fig. 1i) has been considered having a size of 128 $\times$ 128 pixels. This phantom is commonly used to conduct a simulated experiment for a variety of medical imaging applications. It solely consists of ellipses of differing size, shape, orientation, and intensity, superpositioned on one another. To design the mathematical phantom, we set up an imaging geometry using parallel-beam CT scanner and use a circular rotation to acquire 90 projection views uniformly distributed over 360°, where each projection view contains at least 150 measurements. The system matrix a_{ij} is calculated by the geometric calibration method [23] of the pixels. In a practical scenario, it is not possible to obtain the noise-free projections during CT imaging process. Therefore, 10% of Gaussian random noise with zero mean and 0.1 times variance is added to validate the robustness of the proposed method during image reconstruction.

The second test case is a standard real thorax phantom data (shown in Fig. 1ii) acquired from Siemens CT machine. The size of the phantom is 128 $\times$ 128 pixels, similar to other test phantoms considered in this work. The phantom was scanned

by clinical CT scanner having parameter setting as: Tube voltage set at 120 kVp, exposure level set at 100 mAs, the radius of source circle set at 2.868 mm, ray spacing set at 1.2 mm, and the distance between the source and detector band from the center was 408 mm and 816 mm, respectively.

For the evaluation of the performance of the proposed approach in both quantitative and visual manner, different quality measures like peak signal-to-noise ratio (PSNR), the root-mean-square error (RMSE), mean structure similarity index map (MSSIM), and the correlation parameter (CP) have been considered. All the results were performed for 1000 number of iterations.

To implement the other standard methods, the value of different parameters has to be chosen optimally after rigorous experimental works. In the implementation of MLEM + AD [19], the value of kappa was set between 1/100 and 1/300, numbers of iterations of AD were between 3 and 10, and the delta_t value was set to 1/7. In MLEM + TV [21], the value of theta was kept as between 12 and 16°. In MLEM + NLM [22], the degree of filtering was set as 12 and ratio of the window size from 1 to 5. The number of iterations was 100, the value of lambda is set to 0.1, value of alpha is set to 5, and these parameters were also set for the implementation of MLEM + AwTV [24]. The threshold value k of the proposed model was set to be either one or two. Figures 2 and 3 show the output images reconstructed using the proposed model and other models from the four test phantoms. The experimental results show that the proposed model (MLEM + CD) is visually better than all the compared approaches in all the test cases. It is observed that the proposed model (MLEM + CD) in all the four phantoms has shown better results as compared to the other models. The proposed model results are able to preserve the edges and remove the noise.

Figure 4 shows the various results for the quantitative measures of the proposed and compared models. Fig. 4a shows the comparison of the proposed model and other models on the basis of PSNR value and number of iterations. From the convergence plot, it is clearly observed that proposed model has better performance with higher PSNR value as compared to the other models. Higher value of PSNR indicates that the quality of the reconstructed image is improved.

Figure 4b shows the RMSE value and number of iterations. RMSE value for the proposed is smaller as compared to other values. Lesser value of RMSE means that the reconstructed images are more accurate and have less error. In Fig. 4c, d, the convergence graphs of MSSIM and CP are shown. More the MSSIM value is higher (close to unity) indicates that contrast and other details of the images are persevered during reconstruction. It is clearly observable that MSSIM value for the proposed model in both the test cases is higher and hence provides good contrast and detailed images. Tables 1, 2 contain various quantitative parameter values like PSNR, RMSE, MSSIM, and CP. From both the tables, it is clear that in all the parameters the proposed model performs better. Figure 5 shows the profile graphs plotted between the pixel position and pixel intensity values for both the test cases. These graphs depict the accuracy of the model. From the experimental results, it is shown that the proposed model is more accurate or close to the original image as compared to the other models.

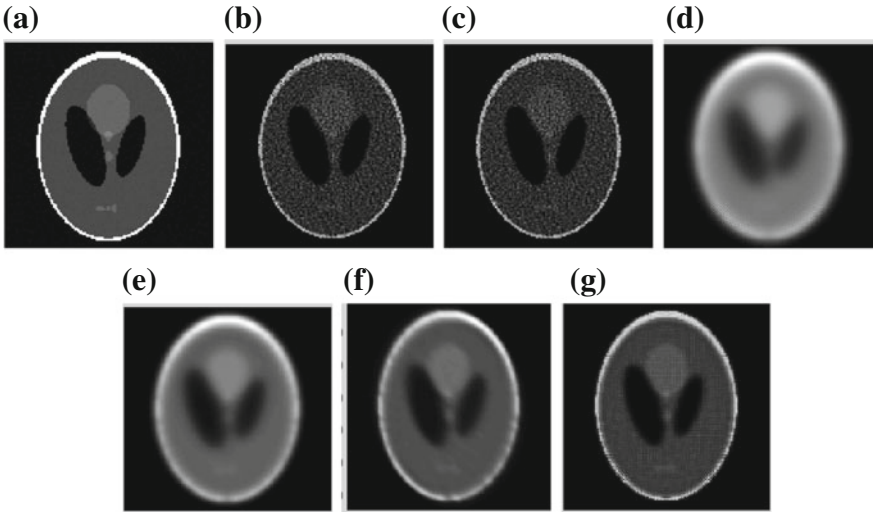


Fig. 2 Reconstructed results of Shepp-Logan head phantom using different methods: **a** original phantom, **b** MLEM, **c** MLEM + AD, **d** MLEM + TV, **e** MLEM + NLM, **f** MLEM + AwTV, and **g** proposed model

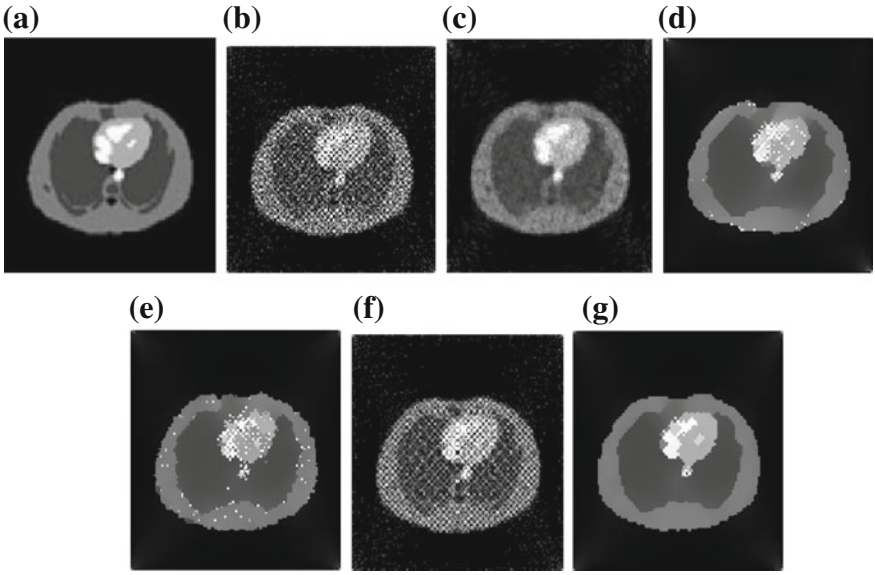


Fig. 3 Reconstructed results of real thorax phantom using different methods: **a** original phantom, **b** MLEM, **c** MLEM + AD, **d** MLEM + TV, **e** MLEM + NLM, **f** MLEM + AwTV, and **g** proposed model

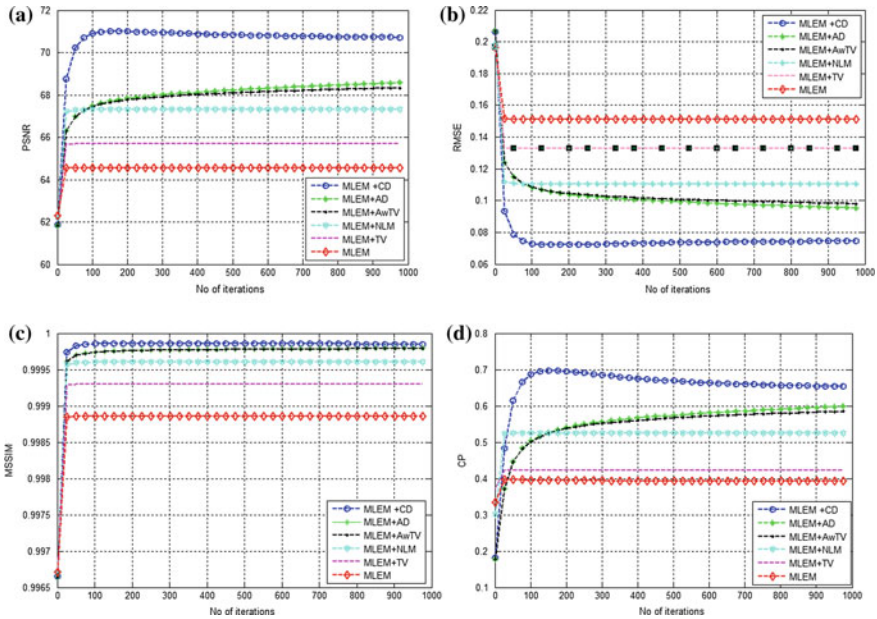


Fig. 4 Convergence plot of iterations vs error metrics: **a** PSNR, **b** RMSE, **c** MSSIM, and **d** CP for modified Shepp-Logan head phantom (test case 1)

Table 1 Quantitative parameters of Shepp-Lagon head phantom for test case 1

	MLEM	MLEM + AD	MLEM + TV	MLEM + NLM	MLEM + ATV	MLEM + CD
PSNR	65.5643	68.5980	65.69991	67.3000	68.3411	71.0055
RMSE	0.0951	0.0951	0.1328	0.1105	0.0980	0.0721
MSSIM	0.9989	0.9998	0.9993	0.9996	0.9998	0.9999
CP	0.3977	0.5994	0.4230	0.5239	0.5852	0.6979

Table 2 Quantitative parameters of standard real thorax test phantom for test case 2

	MLEM	MLEM + AD	MLEM + TV	MLEM + NLM	MLEM + ATV	MLEM + CD
PSNR	62.1421	73.1093	69.5928	71.2615	72.8087	74.5672
RMSE	0.2000	0.0566	0.0848	0.0700	0.0586	0.0478
MSSIM	0.9991	0.9999	0.9995	0.9998	0.9999	0.9999
CP	0.3701	0.8376	0.8815	0.8833	0.8910	0.8794

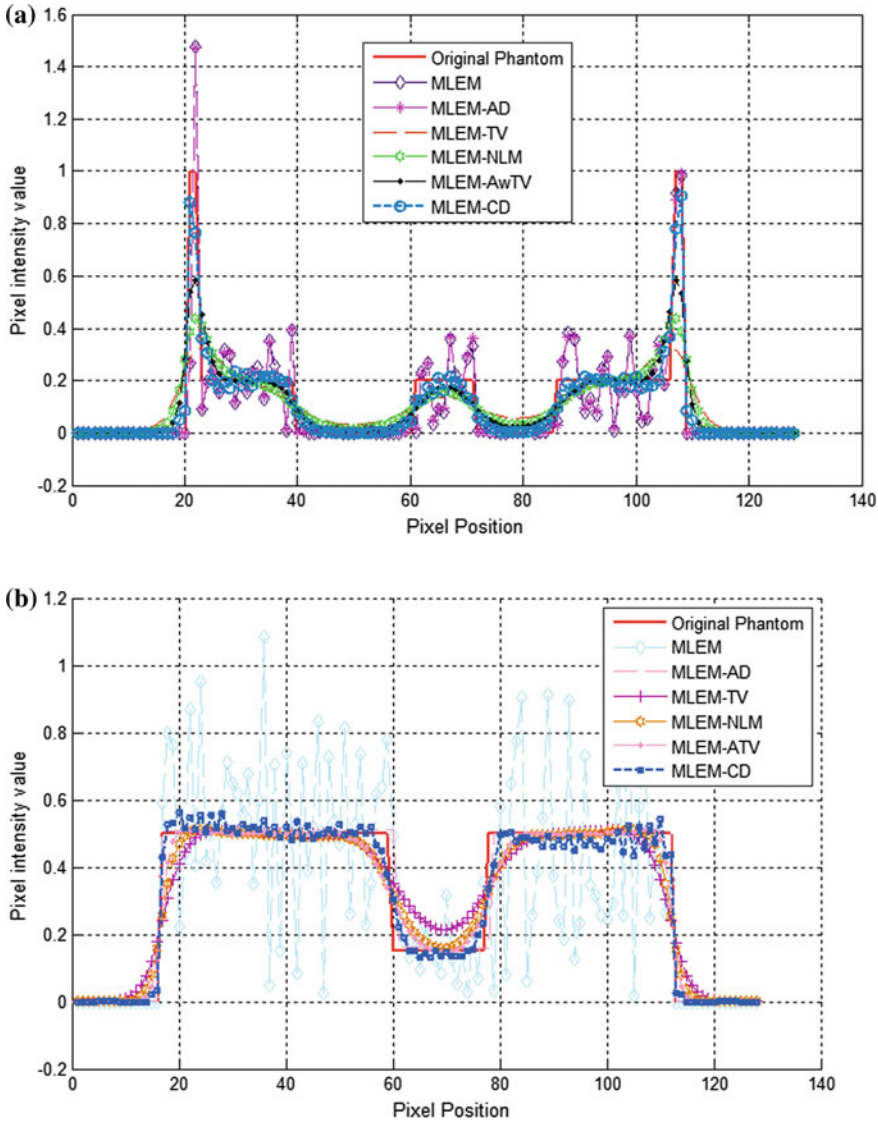


Fig. 5 Profiles along the middle row of **a** Shepp-Logan head phantom and **b** standard real thorax phantom

4 Conclusion

To remove the Gaussian noise as well as ill-posedness issue in the low-dose projection data, a new statistical iterative framework was introduced. The proposed framework is composed of two terms called as data fidelity and a regularization prior, and uses

minimization of energy functional as its foundation. We applied the complex diffusion filter as a prior term with standard MLEM reconstruction algorithm in an iterative manner. The proposed model has been capable of handling Gaussian noise as well as ill-posedness issue effectively. The CD filter is an extension of anisotropic diffusion filter introduced by Perona and Malik (P-M). Finally, we compared the proposed method with other standard procedures in low-dose X-ray CT using qualitative and quantitative analysis. The comparative performance evaluation and the study of the proposed method exhibit its better capability to remove the noise and preserve the minor information.

References

1. Zhang, H., Wang, J., Ma, J., Lu, H., Liang, Z.: Statistical models and regularization strategies in statistical image reconstruction of low-dose X-ray computed tomography: a survey. In: IRIS Laboratory, Department of Radiology, Stony Brook University. arXiv:1412.1732 (2014)
2. Devaney, J.: Filtered backpropagation algorithm for diffraction tomography. Ultrason. Imaging **4**(3), 336–350 (1982). https://doi.org/10.1016/0161-7346(82)90017-7
3. Green, P.G.: Bayesian reconstructions from emission tomography data using a modified EM algorithm. IEEE Trans. Med. Imaging **9**(1), 84–93 (1990). https://doi.org/10.1109/42.52985
4. Vardi, Y., Shepp, L.A., Kaufman, L.: A statistical model for positron emission. J. Am. Stat. Assoc. **80**(389), 8–20 (1985). https://doi.org/10.2307/2288030
5. Hudson, H.M., Larkin, R.S.: Accelerated image reconstruction using ordered subsets of projection data. IEEE Trans. Med. Imaging **13**(4), 601–609 (1994). https://doi.org/10.1109/42.363108
6. Browne, J., De Pierro, A.B.: A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography. IEEE Trans. Med. Imaging **15**(5), 687–699 (1996). https://doi.org/10.1109/42.538946
7. Hsiao, T., Huang, H.M.: An accelerated ordered subsets reconstruction algorithm using an accelerating power factor for emission tomography. Phys. Med. Biol. **55**(3), 599 (2010). https://doi.org/10.1088/0031-9155/55/3/003
8. Lee, S., Kwon, H., Cho, J.: The detection of focal liver lesions using abdominal CT: a comparison of image quality between adaptive statistical iterative reconstruction V and adaptive statistical iterative reconstruction. Acad. Radiol. **23**(12), 1532–1538 (2016). https://doi.org/10.1016/j.acra.2016.08.013
9. Gordon, R., Bender, R., Herman, G.T.: Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. J. Theor. Biol. **29**(3), 471–476 (1970). https://doi.org/10.1016/0022-5193(70)90109-8
10. Geyer, L.L., Schoepf, U.G., Meinel, F.G., Nance Jr., J.W.: State of the art: iterative CT reconstruction techniques. Radiology **276**(2), 339–357 (2015). https://doi.org/10.1148/radiol.2015132766
11. Gordic, S., Desbiolles, L., Stolzmann, P., Gantner, L.: Advanced modelled iterative reconstruction for abdominal CT: qualitative and quantitative evaluation. Clin. Radiol. **69**(12), e497–e504 (2014). https://doi.org/10.1016/j.crad.2014.08.012
12. Levitan, E., Herman, G.T.: A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography. IEEE Trans. Med. Imaging **6**(3), 185–192 (1987). https://doi.org/10.1109/TMI.1987.4307826
13. Denisova, N.V.: Bayesian reconstruction in SPECT with entropy prior and iterative statistical regularization. IEEE Trans. Nucl. Sci. **51**(1), 136–141 (2004). https://doi.org/10.1109/TNS.2003.823059

14. Perona, P., Malik, J.: Scale-space and edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990). https://doi.org/10.1109/34.56205
15. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based. Phys. D **60**(1–4), 259–268 (1992). https://doi.org/10.1016/0167-2789(92)90242-F
16. Ling, H., Bovik, A.C.: Smoothing low-SNR molecular image via anisotropic median-diffusion. IEEE Trans. Med. Imaging **21**(4), 377–384 (2002). https://doi.org/10.1109/TMI.2002.1000261
17. Gilboa, G., Zeevi, Y.Y., Sochen, N.A.: Complex diffusion processes for image filtering. In: Third International Conference on Scale-Space and Morphology in Computer Vision, Scale-Space 2001, Vancouver, Canada (2001). https://doi.org/10.1007/3-540-47778-0
18. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. IEEE Comput. Vis. Pattern Recogn. **2**, 60–65 (2005). https://doi.org/10.1109/CVPR.2005.38
19. Yan, J.H.: Investigation of positron emission tomography image reconstruction. Huazhong University of Science & Technology, Wuhan, China (2007)
20. He, Q., Huang, L.: Penalized maximum likelihood algorithm for positron emission tomography by using anisotropic median-diffusion. Math. Probl. Eng. Hindawi **2014**, 1–7 (2014). https://doi.org/10.1155/2014/491239
21. Chávez-Rivera, L.B., Ortega-Máynez, L., Mejía, J., Mederos, B.: ML-EM reconstruction model including total variation for low dose PET high resolution data. In: Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2015 IEEE, San Diego (2015). https://doi.org/10.1109/nssmic.2015.7582221
22. Zhang, H., Ma, J., Liu, Y., Han, H., Li, L., Wang, J., Liang, Z.: Nonlocal means-based regularizations for statistical CT reconstruction. In Proceedings of SPIE 9033, Medical Imaging 2014: Physics of Medical Imaging, 903337, San Diego (2014). https://doi.org/10.1117/12.2043949
23. Tiwari, S., Srivastava, R.: A hybrid-cascaded iterative framework for positron emission tomography and single-photon emission computed tomography image reconstruction. J. Med. Imaging Health Inf. **6**(8), 1–12 (2016). https://doi.org/10.1166/jmihi.2016.1779
24. Liu, Y., Ma, J., Fan, Y., Liang, Z.: Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction. Phys. Med. Biol. **57**(23), 7923 (2012). https://doi.org/10.1088/0031-9155/57/23/7923

MatConvNet-Based Fast Method for Cervical MR Images Classification

Gyanendra K. Verma, Jagdeep S. Lather and Ankur Kaushal

Abstract The deep convolutional neural network (CNN) has been successfully used to obtain high-level representation in various applications of computer vision problems. However, in the field of medical imaging there are not sufficient images available to train a deep CNN. Therefore, we have used a pre-trained deep CNN model for classification of cervical cancer MR images. In this paper, we have proposed MatConvNet-based CNN model to extract features from pre-trained CNN for classification. The vgg-f architecture is deployed to extract the image features. We have evaluated our proposed system with benchmark cervical cancer database obtained from Tumor Cancer Imaging Archive (TCIA). We got the promising result with 98.9% accuracy that is beyond the methods reported in the literature.

Keywords Convolutional neural network · Cervical cancer detection · Deep learning

1 Introduction

Automatic classification of medical images particularly cancer cells is a very challenging task because of high interclass variability and low interclass distance. Over the most recent two years, deep convolutional network (DCN) has performed beyond

G. K. Verma (✉)
Department of Computer Engineering, National Institute of Technology Kurukshetra, Kurukshetra 136119, India
e-mail: gyanendra@nitkkr.ac.in

J. S. Lather
Department of Electrical Engineering, National Institute of Technology Kurukshetra, Kurukshetra 136119, India
e-mail: jslather@nitkkr.ac.in

A. Kaushal
School of Biomedical Engineering, National Institute of Technology Kurukshetra, Kurukshetra 136119, India
e-mail: ankur.vicky999@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

N. K. Verma and A. K. Ghosh (eds.), *Computational Intelligence: Theories, Applications and Future Directions—Volume II*, Advances in Intelligent Systems and Computing 799, https://doi.org/10.1007/978-981-13-1135-2_51

state-of-the-art methods in numerous computer vision applications, i.e., [1, 2]. While convolutional networks have existed for quite a while [3], their application was constrained due to the limited size of the accessible training sets and the span of the considered networks. According to American Cancer Society, there are about 13,240 new cases of invasive cervical cancer and about 4,170 people will die from this cancer in the USA for 2018 [4]. This cancer is diagnosed by cervical cancer screening test (Pap smear) in which cells taken from cervix undergone a microscopic examination. However, this test is highly dependent on the cytopathologists' proficiency which has the sensitivity of 53.4–64.2% [5].

Clinical staging has many deficiencies which include parametrial and pelvic sidewall invasion, and size of endocervical tumours for cervical cancer. Magnetic resonance (MR) imaging is noticed as the most preferable modality for the pre-treatment work-up of cervical cancer because it has superior soft tissue resolution and multiplanar capability and it also removes the use of invasive examination such as cystoscopy and proctoscopy. MR imaging is also used to calculate the lesion volume and lymph node involvement. Generally, T2-weighted MR imaging is used to evaluate cervical carcinoma and lymph nodes and T1-weighted imaging in which contrast is enhanced by gadolinium that is used for small tumor detection [1].

Automatic segmentation of cervical cancer in MR imaging has many applications in cervical cancer diagnosis and therapy. We use a pre-trained CNN model to segment cervical cancer automatically. So in our experiment, we take the MR images of uterine cervix to diagnose the tumor by automatic segmentation with the help of Visual Geometry Group (VGG) model which is a Convolutional Neural Network (CNN) trained on ImageNet dataset.

The notable contributions of this work can be given as

- This is the primary work (to the best of our knowledge) finished with cervical MR images with pre-trained CNN as past work depends on pap-spread cell method.
- The computation time is low in contrast with existing frameworks as past studies were actualized on caffe model that takes much time compare to MatConvNet (MATLAB) without GPU.
- We have utilized MatConvNet model that does not require GPU. All the experiments are performed with Intel Core i7-4790 CPU with 8 GB Ram.

Rest of the paper is organized as follows. Related work is given under Sect. 2. Methodology is discussed under Sect. 3 followed by experimentation and result analysis under Sect. 4. Finally, concluding remarks are given under Sect. 5.

2 Related Works

Deep learning algorithms, particularly convolutional neural networks (CNN), have been used in various applications of computer vision. Many researchers started using CNN with pre-trained model of ImageNet. The application of CNN algorithms includes the image segmentation and object detection. Application of deep

learning in medial image classification and segmentation removed the barriers of cancer diagnosing. The use of CNNs and other deep learning algorithms explored over the past two–three years are limited in 2014 to approximately 50 papers in 2015 and 500 in 2016.

Spanhol et al. [6] conducted a set of experiments with BreaKHis dataset using pre-trained model based on AlexNet [7] CNN architecture. AlexNet is a CNN model trained on huge ImageNet database for classifying color images of natural objects and scenes. The authors have used above model to classify histopathological breast cancer images. Eventually, the authors fine-tuned the AlexNet architecture and its layers of classification while leaving feature layer unchanged. They claimed good accuracy with BreaKHis dataset if compared to the existing methods reported in the literature.

Xu et al. [8] proposed a multimodal deep neural network for diagnosing cervical dysplasia. The authors collected highly heterogeneous data from the patients screening visit and expand conventional CNN architecture to fully connected layers. Here, multimodal framework is an end-to-end deep network which extracts features from images and non-image modalities [8] and sensitivity for the cervical dysplasia diagnosis is 87.33–90% specificity on large dataset which is a very good result on limited information or dataset.

Cernazanu et al. [9] conducted a study in which authors take the X-ray images, and through segmentation, they differentiate the bone tissue from background tissue; there are many segmentation techniques available but most of them are based on hand crafted features. In this article, authors proposed a new segmentation technique using CNN model. At present, CNN is very preferred segmentation algorithm in the field of medical image analysis, generally X-ray has large dimension and for reducing the training time authors only select certain area of maximum interest from the whole image. Here, neural network works as a pixel classifier which label each pixel from raw pixel values in a square area; thus, network classifies the image region in two classes (bone, non-bone). They have also conducted experiment for extracting an accurate contour of the bones from the raw X-ray images. Authors introduced CNN architecture with modified max-pooling layers; the main role of these layers is to extract the important features at a specific time. Thus, the result of bone segmentation is remarkable good and also with improved accuracy.

3 Methodology

Practically, few people train the whole CNN network from scratch because it needs a huge training dataset for medical images. So, CNN architectures ResNet, VGG, AlexNet, UNet were the most utilized pre-trained deep learning models. Transfer learning and fine-tuning of these CNN models for classification and segmentation of medical images are drawing attention of doctors and researchers. However, there are also some unique problems of falsely labeled images and low-quality dataset in training these deep learning CNN models. Few recent studies in the field of

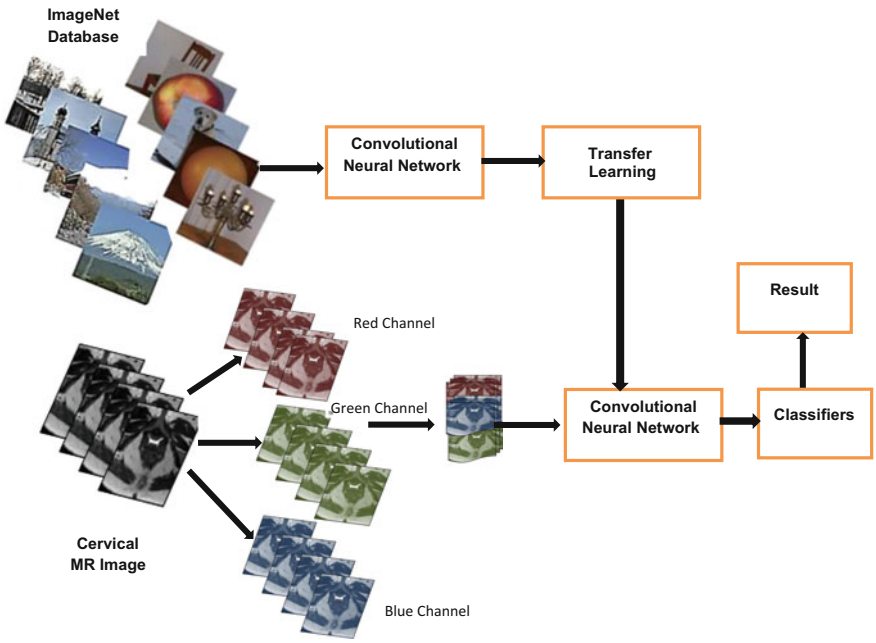


Fig. 1 System architecture

medical imaging that uses deep learning architecture are [2, 3, 7]. In [7], ImageNet non-medical trained model has been used for classification of medical images. By same inspiration, we use pre-trained CNN model Visual Geometry Group (VGG) in our cervical images feature extraction and classification. A system architecture is depicted in Fig. 1.

3.1 VGG-F Architecture

We have executed CNN using MatConvNet [10], a MATLAB tool stash completing best in class convolutional neural networks (CNNs) without GPU for PC vision applications. The VGG architecture was introduced by Simonyan and Zisserman in the year 2014. It is popular due to its simplicity, using 33 convolutional layers which are stacked on top of each other. It contains max-pooling layers which help in reducing the volume size, two fully connected layers, each having 4096 nodes and a softmax classifier. Interesting thing is that we can modify each and every layer of the network according to our challenge. We can modify convolutional, rectified linear unit (ReLU) or feature classification layers. However, there are also some drawbacks of this network: (1) It takes a lot of time in training and also tedious to train network. (2) The network architecture weight is itself quite large (in terms of disk/bandwidth). The actual size of this network is over 533 MB. In our work, we utilize VGG-F a

modified VGG pre-trained network [11], the architecture of VGG-F incorporates five convolutional layers and three totally connected layers with different parameters. VGG-F uses dropout regularization in the completely associated layer and applies local reaction standardization.

3.2 Feature Extraction Using VGG-F Model

Convolutional neural networks use features to classify images. The network learns these features itself during the training process. The convolutional layers of a network yields multiple 2-D arrays. Each array (or channel) corresponds to a filter applied to the layer input. The channels output by fully connected layers correspond to high-level combinations of the features learned by earlier layers.

Most important characteristics of this network are learning hierarchical layers of different features, and some middle layers describe the edges orientation in the image while other higher layers describe parts of object or complete object. Accuracy of the deep learning models is dependent upon the large training dataset but in medical imaging such labeled large dataset is very difficult to obtain [12].

We have deployed three different classifiers namely support vector machine (SVM), ensemble tree, and K-nearest neighbors rather using softmax function to classify cervical MR images. Detail descriptions of classifiers are given under Sect. 4.3.

4 Experimental Setup and Result Analysis

4.1 Database

In our experiment, we take the open access MR data of cervical from Cancer Imaging Archive (TCIA) [13]. The database contains 19,135 magnetic resonance images in dicom format obtained from 54 patients and the same is available freely to download for scientific and educational research purpose. Sample MR images are given in Fig. 2.

4.2 Evaluation Matrices

We have utilized distinctive performance measurements such as accuracy, True Positive Rate (TPR), False Positive Rate (FPR) and Area Under Curve (AUC) for evaluation of the system.

Recall is the extent of number of true positives detected to the aggregate number of positive inputs, and it is also termed as TPR while, precision is the extent of

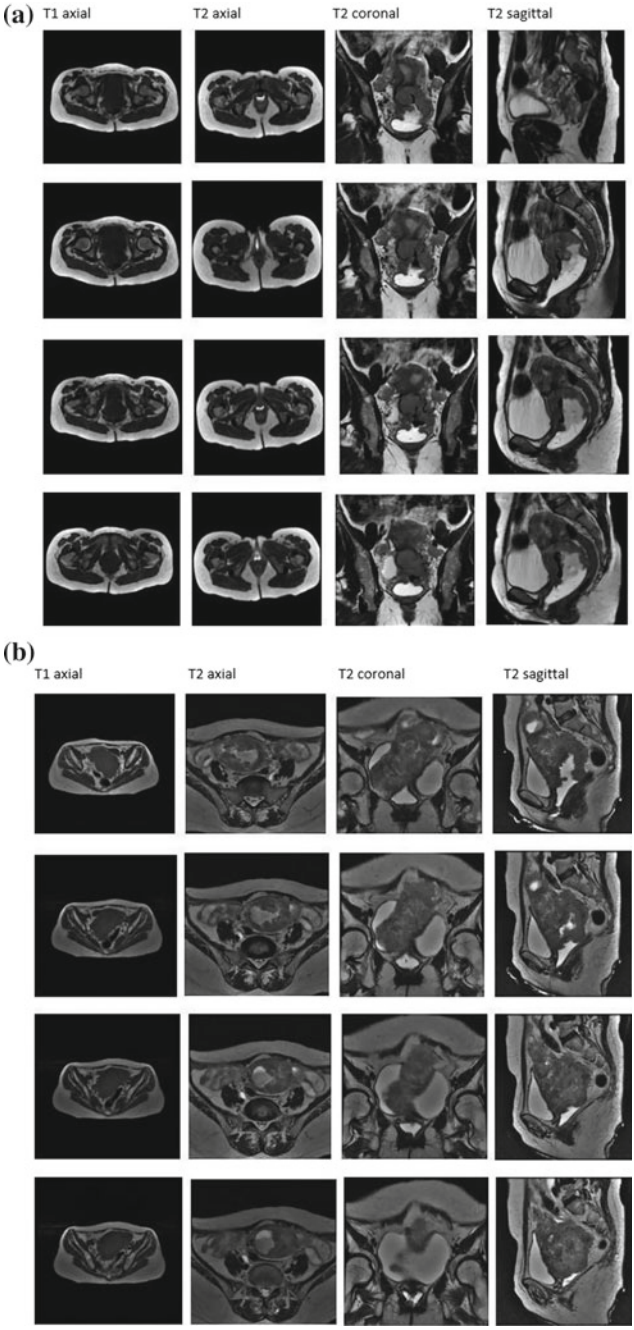


Fig. 2 Cervical cancer images of TCIA **a** positive, **b** negative sample

number of true positives to the number of positive calls; it is also termed as Positive Predictive Value (PPV). Receiver Operating Characteristic curve (ROC curve) is a plot between precision and recall as given in Eqs. 1 and 2, respectively.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

4.3 Implementation Details

We use MATLAB 2016a with hardware plan as Intel Core i7-4790 CPU @ 3.60 GHz (8 GB RAM) 64-bit operating system for all experimentation. We have used MatConvNet, a MatLab based tool that can perform image analysis with CPU using vgg-f pre-trained model. In this way, we have used significant framework for CNN feature extraction.

We have utilized distinctive classifiers for the classification of cervical cancer from MR images. We have utilized three state-of-the-art classifiers namely Support Vector Machine (SVM), Ensemble Tree, and K-Nearest Neighbors (KNN) for classification purpose. The high-level features extracted from DCNN are fed to above classifiers to acquire a class label (positive and negative) for each input data. SVM is a supervised learning classifier that can be utilized for both linear and nonlinear classification. We have employed different variants of SVM with linear kernel function and the best one is reported as result. Additionally, we have employed quadratic and cubic SVM with quadratic and cubic kernel functions and automated kernel scale. The medium Gaussian SVM utilizes Gaussian kernel function with kernel scale as 32.

KNN is an instance-based technique broadly utilized in multiple fields, for example, Medical Imaging, Pattern Recognition, Information Retrieval. We have utilized cosine and weighted KNN each with 10 numbers of neighbors. Cosine KNN utilizes cosine distance metric with equivalent distance weights. In Weighted KNN, the closer neighbors are weighted more heavily than, the farther ones, using the distance-weighted function. We have used weighted KNN with K = 1 for experimentation. Ensemble classifiers are likewise utilized; that uses numerous weak learners for classification. Ensemble boosted tree is utilized with adaboost ensemble method, decision tree learner, 20 number of splits, 30 number of learners, and 0.1 learning rate. Ensemble bagged tree is also used with bag ensemble technique, decision tree learner type, 30 number of learners, 20 number of splits, and 0.1 learning rate.

4.4 Results and Discussions

The system accuracy with three different classifiers are given in Table 1. As we can see from Table 1, the highest accuracy is achieved with KNN classifier that is 98.9%. The performance of SVM classifier is also recognizable with 97.7%. The performance of the system is given in terms of TPR, FPR and AUC. The ROC curve and related confusion matrix for each classifier, i.e., SVM, ensemble tree, and KNN are depicted under Fig. 3. We observe that DCNN features alongside learning approach for classification improve the system performance. The verification step provides high true positive rates along with low false positive rates.

Additionally, the highest true positive rate is 0.99 with KNN while others classifiers have true positive predictive value in the range of 0.89–0.97 that is likewise viewed as great outcomes. However, the most noteworthy accuracy is 98.9% acquired with weighted KNN classifier. The computation time and prediction speed for different classifiers are given in Table 2.

4.5 Performance Comparison

Performance comparison with existing studies is given in Table 3. Hyeon et al. [14] used VGG-16 pre-trained CNN model in the ImageNet large-scale visual recognition challenge as a feature vector extractor in microscopic images and collected 71,344 Pap smear microscopic images and their interpretations from Seegene Medical Foundation. They divide the dataset into 80% training set and 20% test set and train all classifiers using only the training set and test them using only the test set. The result of classification performance is maximum for SVM classifier with 78%.

Taha et al. [15] extracted low- and high-level features from different layers of convolutional neural network and evaluated the performance of the system in order to evaluate the CNN as a feature extractor and its adequacy in the recognition on the predetermined database. The primary concentration in the analysis is the quality, as far as characterization control, of the extracted features from the CNN. AlexNet, which comprises millions of non-medical images of thousand classes, was utilized to train the model with pap-spread cell images.

Bora et al. [16] had chosen AlexNet pre-trained CNN model. Experiments were performed on generated database as well as on Pap smear database with 1611 images.

Table 1 System performance with TCIA database

Classifier	Accuracy (%)	TPR	FPR	AUC
SVM	97.7	0.97	0.02	0.99
Ensemble tree	89.8	0.89	0.09	0.88
KNN	98.9	0.99	0.01	0.99

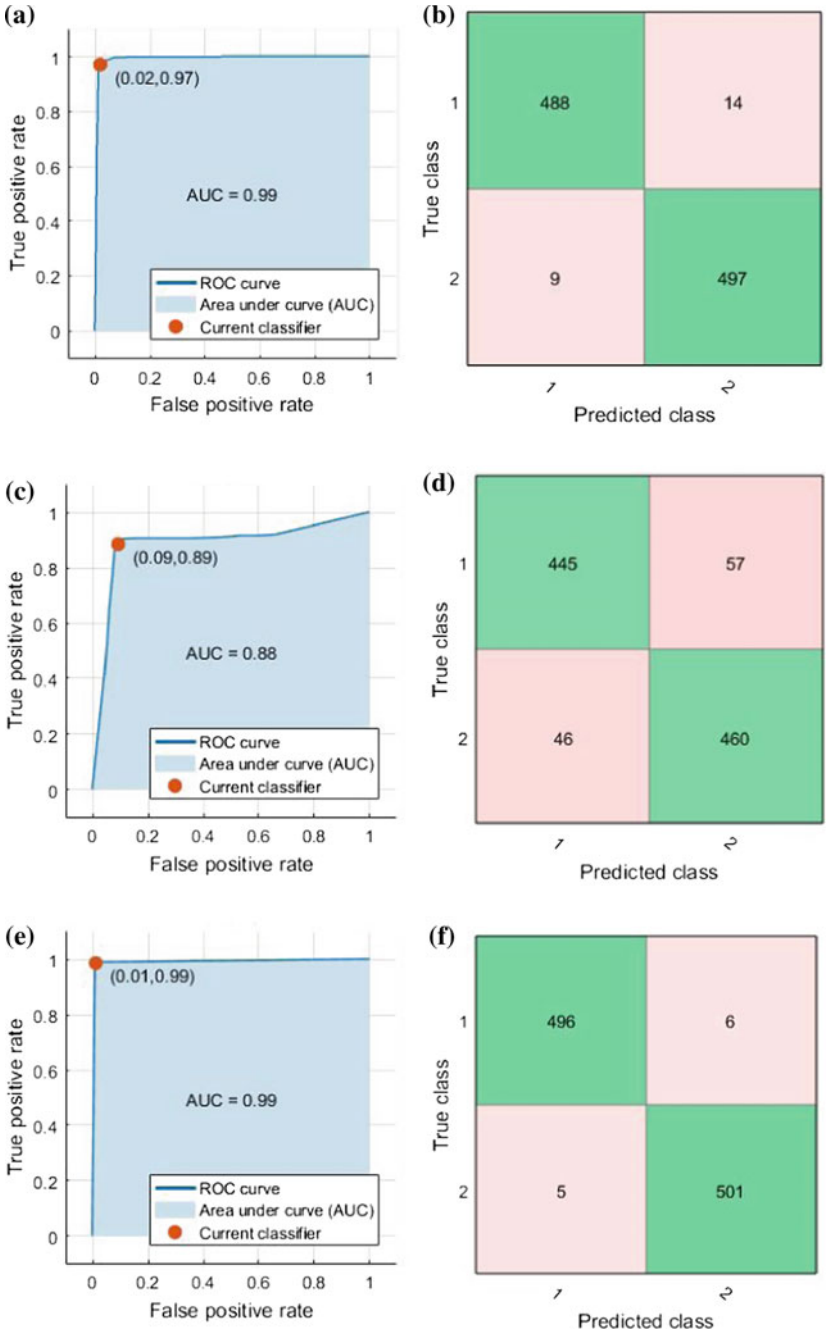


Fig. 3 ROC curve and confusion matrices for different classifiers (**a, c, e** ROC curves; **b, d, f** confusion matrices) (**a, b** SVM; **c, d** ensemble tree; **e, f** KNN)

Table 2 Training time and prediction speed for different classifiers

Classifiers	Training time (s)	Prediction speed (observations/s)
SVM	3.476	~1600
KNN	3.404	~1200
Ensemble tree	8.387	~1800

Table 3 Accuracy comparison with existing studies

Reference	Methods	Database	Year	Accuracy (%)
Hyeon et al. [14]	CNN (vgg-16)	Pap smear microscopic images	2016	78.0
Taha et al. [15]	AlexNet CNN	Pap smear cell image	2017	98.37
Bora et al. [16]	AlexNet CNN	Pap smear cell image	2017	95.0
This study	MatConvNet (vgg-f)	TCIA MR images	2017	98.9

They achieved accuracy in the range of 90–95% for two- and three-class classification.

5 Conclusion

This paper presents cervical cancer recognition based on classification of MR images using MatConvNet-based CNN architecture. The distinct CNN features are extracted using transfer learning of pre-trained vgg-f model. The proposed method achieves promising performance compared to other biomedical segmentation methods reported in the literature. Our system is very fast and accurate with elastic deformations as it only needs few annotated images and has a very short training time without GPU.

In future, we will experiment with other database for different cancer category such as prostate, brain tumor.

References

1. Nicolet, V., Carignan, L., Bourdon, F., Prosmanne, O.: MR imaging of cervical carcinoma: a practical staging approach. Radiographics **20**(6), 1539–1549 (2000)
2. Bar, Y., Diamant, I., Wolf, L., Lieberman, S., Konen, E., Greenspan, H.: Chest pathology detection using deep learning with non-medical training. In: 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), pp. 294–297. IEEE, Apr 2015

3. Samala, R.K., Chan, H.P., Hadjiiski, L., Cha, K., Helvie, M.A.: Deep-learning convolution neural network for computer-aided detection of micro calcifications in digital breast tomosynthesis. In: International of Society for Optics and Photonics, SPIE Medical Imaging, 97850Y-97850Y (2016)
4. American Cancer Society. https://www.cancer.org/cancer/cervical-cancer/about/key-statistics.html. 10 July 2017
5. Kahng, J., Lee, H.J.: Clinical efficacy of HPV DNA chip test in the era of HPV vaccination: 1,211 cases, a single institution study. Korean J. Lab. Med. **28**(1), 70–78 (2008)
6. Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L.: Breast cancer histopathological image classification using convolutional neural networks. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 2560–2567. IEEE, July 2016
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
8. Xu, T., Zhang, H., Huang, X., Zhang, S., Metaxas, D.N.: Multimodal deep learning for cervical dysplasia diagnosis. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 115–123. Springer International Publishing, Oct 2016
9. Cernazanu-Glavan, C., Holban, S.: Segmentation of bone structure in X-ray images using convolutional neural network. Adv. Electr. Comput. Eng **13**(1), 87–94 (2013)
10. MatConvNethttp: www.vlfeat.org/matconvnet/pretrained
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv:1409.1556
12. Gentav, A., Aksoy, S., Öner, S.: Unsupervised segmentation and classification of cervical cell images. Pattern Recogn. **45**(12), 4151–4168 (2012)
13. TCIA Database. https://wiki.cancerimagingarchive.net/display/Public/TCGA-CESC
14. Hyeon, J., Choi, H.J., Lee, B.D., Lee, K.N.: Diagnosing cervical cell images using pre-trained convolutional neural network as feature extractor. In 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 390–393. IEEE, Feb 2017
15. Taha, B., Dias, J., Werghi, N.: Classification of cervical-cancer using pap-smear images: a convolutional neural network approach. In: Annual Conference on Medical Image Understanding and Analysis, pp. 261–272. Springer, Cham, July 2017
16. Bora, K., Chowdhury, M., Mahanta, L.B., Kundu, M.K., Das, A.K.: Pap smear image classification using convolutional neural network. In: Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, p. 55. ACM, Dec 2016